

Multi-Lens Cameras in appleseed



Bachelor Thesis
23rd August 2021

Jan Willi
17-923-053

Supervisors:
Prof. Dr. Renato Pajarola
Lars Zawallich

Visualization and MultiMedia Lab
Department of Informatics
University of Zürich



University of
Zurich^{UZH}



Abstract

Rendering engines like appleseed¹ use camera simulations for image formation. Oftentimes however, such camera models are simplifications of realistic cameras that compromise true realism for rendering efficiency. In this thesis, appleseed's camera models are extended by a physically-based multi-lens camera based on scientific literature that features realistic lens phenomena. The implementation is then tested and compared to simpler camera models. The results show that the implemented model is capable of capturing a multitude of optical phenomena of real cameras and that these effects achieve realism better than existing camera models. While a certain rendering deficiency is found, it is shown that this overhead solely depends on the used lens's complexity and, under the right conditions, accounts for a fraction of the total rendering time.

¹appleseed is an open source, physically-based global illumination rendering engine designed for animation and visual effects.

Zusammenfassung

Rendering Engines wie appleseed² setzen Simulationen von Kameras ein, um Bilder zu generieren. Meist sind solche Kameramodelle allerdings Vereinfachungen von realistischen Kameras, die wahren Realismus für Rendering-Geschwindigkeit opfern. In dieser Arbeit werden appleseed's Kameramodelle um ein physikalisch korrektes Kameramodell ergänzt, das auf wissenschaftlichen Publikationen basiert und fähig ist, Kameraphänomene realitätsgetreu abzubilden. Die Implementation wird danach auf ihre Richtigkeit überprüft und mit einfacheren Kameramodellen verglichen. Die Resultate zeigen, dass das implementierte Modell in der Lage ist, eine Vielzahl an optischen Phänomenen von realistischen Kameras einzufangen, die zu verbessertem Fotorealismus im Vergleich zu einfacheren Kameras führen. Obwohl gewisse Geschwindigkeitseinbussen in Kauf genommen werden müssen, kann gezeigt werden, dass der Rendering-Mehraufwand alleinig von der Komplexität des Objektivs abhängt. Weiter nimmt der Mehraufwand, unter geeigneten Bedingungen, nur einen Bruchteil der Gesamtrenderzeit in Anspruch.

²appleseed ist eine Open Source Rendering Engine konzipiert für Animation und visuelle Effekte.

Contents

Abstract	iii
Zusammenfassung	iv
1 Introduction	1
2 Background and Related Work	3
2.1 Background Knowledge	3
2.1.1 Camera Models	3
2.1.2 Aberrations	7
2.1.3 Bokeh Effect	9
2.1.4 Other Effects	10
2.2 Related Work	11
2.2.1 Ray Simulation Camera Models	11
2.2.2 Polynomial Camera Models	12
3 Model	13
3.1 Model Comparison	13
3.2 Model Selection	15
4 Implementation	16
4.1 Lens Specification	16
4.2 Pupil Calculation	17
4.3 Sequential Ray Tracing Through the Lens	18
4.3.1 Pupil Sampling	18
4.3.2 Intersection Validation	19
4.3.3 Refraction	19
5 Results	20
5.1 Achieved Effects	20
5.1.1 Variable Lens Properties	20
5.1.2 Aberrations	20
5.1.3 Vignetting	23
5.2 Unachieved Effects	24
6 Analysis	25
6.1 Feature Range	25
6.2 Realism	25
6.3 Performance	28
6.3.1 Multi-Lens Overhead	29
6.3.2 Custom Aperture Shape	31
6.3.3 Ray Differentials	32
7 Conclusion and Future Work	33

Contents

Bibliography	35
Appendix A Lens Designs	38
Appendix B Performance Measurements	39
Appendix C Performance Renders	42

1 Introduction

Photorealistic image quality is the flagship of many rendering engines. While this mostly concerns the transportation of light and the interaction of light with materials, a suitable camera model is of importance as well, but tends to be neglected [KMH95]. Instead, many rendering engines settle for simpler approximations of complex models, who allow for photographic effects to a certain degree, like depth of field or motion blur. As they are modelled in an optically perfect way and omit imperfections that arise in realistic lens systems of multiple lens elements, many phenomena that constitute photorealism fall short. To take these effects into account, light transport between the scene and the film has to be computed as if a real camera was placed in between [SDHL11]. In this thesis, such a physically-based multi-lens camera model is implemented in the open source rendering engine *appleseed* [Bea+19]. Although not commonly featured in production rendering engines due to the prejudices of slowness and inefficiency, an accurate physically-based model offers advantages in various situations. First and foremost, it fills the gap of the shortcomings of simpler camera models, such as the generation of optical aberrations, realistic vignetting or bokeh effects. Only the inclusion of these phenomena enables the imitation of real-life photography. Beyond that, computer-generated imagery finds extensive use in areas like augmented reality, virtual reality, or the film industry. Whenever real imagery is merged with digital generations, the two must not be distinguishable, which can only be achieved by rendering with a camera similar to the one used for filming [SDHL11]. Additionally, the simulation of a real camera and its correct behavior facilitates its use, as most people are familiar with basic camera operation and have a sense for the image outcome based on their camera setup and settings [KMH95].

Especially in the current century, a number of physically-based camera models have been contributed to the literature. For many, the work of Kolb et al. from 1995 [KMH95] serves as a foundation to build on. They were the first to propose all requirements to realistically compute light transport through a real lens. Utilizing prescriptions of real lenses, they let light rays interact with the lens in a physically accurate way. In later advancements of this ray simulation approach, performance, range of effects, and adjustability were improved, but the core idea was kept [Wu+10; SDHL11; WZHX13]. Only in 2012, a new way of camera simulation emerged. Hullin et al. [HHH12] approximate the analytical light transport solution by a polynomial system, thereby omitting ray tracing through the lens. In a number of follow-up papers, improvements were suggested, especially concerning the construction of the polynomial and the approximation error that determines the image quality [HD14; SHD16; ZZ17b; ZZ17a].

The goal of this thesis is to contribute a multi-lens camera model to *appleseed*'s open source repository that is able to render physically accurate imagery including camera-induced effects. Besides full functionality, it should convey the feeling of a real camera and offer adjustability of focal length, focus distance and aperture size and shape, similar to what a photographer can do with his camera. To start off, different camera models relevant to computer graphics are presented and their characteristics and features are reasoned with reference to the optical foundations thereof. Following that, realistic camera imperfections, so-called aberrations, are discussed, which account for a large part of the added realism of the multi-lens camera over simpler camera models. In addition to aberrations, other distinct phenomena such as bokeh, lens flare, or vignetting are talked about. In the second part of Chapter 2, a literature overview is presented, showcasing the current state of modeling realistic cameras. It can roughly be divided into ray simulation models based on Kolb et al., physically accurate calculations of how a ray moves through the lens, and polynomial models following Hullin et al., approximating the behavior of the lens using polynomials. In Chapter 3, different concrete models of both literature paths are compared on various aspects. While most papers remain relatively silent about performance, especially comparing ray simulation to polynomial approximation, some usable hints can be found nonetheless. On the topic of performance, different

1 Introduction

sampling methods are assessed, which play an important role for efficiency when tracing rays through the lens. When it comes to the range of supported phenomena, much clearer distinctions can be made, although most models stay within similar bounds. Finally, under consideration of said factors and the suitability for appleseed's environment and codebase, a choice for implementation is made, which is then described in detail in Chapter 4. Chapter 5 demonstrates the correctness of the implementation by presenting renders that exhibit the previously discussed phenomena. The last chapter is then dedicated to an analysis of the implementation. Besides supported effects and image quality, an in-depth performance evaluation is conducted, relating the implemented model to other camera models of appleseed, like the pinhole or thin lens camera.

2 Background and Related Work

This chapter first presents relevant background knowledge about camera models and optical phenomena thereof. While many concepts originate from optical sciences, the focus lies on computer graphics. Optical background is provided where necessary, but not expanded on greatly. The interested reader is referred to book such as *Introduction to Optics* [PP93], *Modern Optical Engineering* [Smi00] or *Optics* [Hec17]. In a second segment, the current state of research about realistic cameras in rendering is presented, differentiating between ray tracing through lens systems and polynomial approximations of it.

2.1 Background Knowledge

A rendering engine can utilize different camera models when transforming a 3D model into a 2D image. Depending on the needs, such cameras can range from primitive pinhole to realistic photographic lens models. With increasing realism and digital modeling of real physics, a multitude of optical effects known from real photography is introduced in the rendered images [Wei91].

2.1.1 Camera Models

All camera models that see use in computer graphics share some common traits. Their task is to capture light from the object space onto a two-dimensional image space. Despite different internals, all models project objects onto an image plane or film plane, in digital cameras a sensor [Bar+03].

Pinhole Camera

The pinhole camera, depicted in Figure 2.1, is the most frequently used model in computer graphics [HSS97; SDHL11]. Its concept dates back to the Middle Ages [You89]. Later, Leonardo da Vinci used the pinhole model to allegedly solve the problem of why objects cast different images at different distances onto a screen [Thr94]. Also known as camera obscura, this camera is a box with a single opening, through which light rays can fall into the box. The hole has no diameter, therefore only a single ray of light from each point in the scene can fall into the box. On the opposite wall inside the camera, an upside-down image is projected [HSS97; Bar+03].

While it is possible to build a working camera obscura, it is not very practical as the small hole only allows for very little light to pass and requires long exposure times. Its simplicity, however, make this model very attractive in terms of efficiency for rendering engines [ZZ17b].

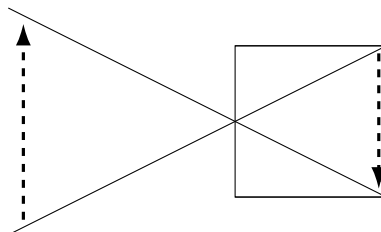


Figure 2.1: The pinhole camera projects the scene upside-down onto the image plane inside the box.

Thin Lens Model

The pinhole camera with no diameter is a theoretical concept and only works in the realm of computers. In reality, every physically built pinhole camera has a finite opening, hereafter called aperture, that lets many light rays pass. The simplest of such finite aperture models is the thin lens approximation that finds use in optics and lens design [HSS97]. In essence, it consists of a single spherical lens that is assumed to have no thickness [Smi00; Bar+03]. That way, an incoming ray of light is refracted only once by the lens on the so-called principal plane and otherwise moves in a straight line [BW99; HSS97].

The schematics of the thin lens model are visualized in Figure 2.2. Light from a point P in object space is focused in a single point P' in image space. Additionally, two focal points F and F' exist. All light rays coming from object space that are parallel to the optical axis meet at F' after being refracted by the lens, while analogously, parallel image space rays meet at F [HSS97; Bar+03]. The distance f between a focal point and the principal plane is known as the focal length [Smi00]. If the indices of refraction on both sides of the lens are the same, the focal lengths are equal [HSS97]. The main law around image formation in a thin lens is the thin lens equation

$$\frac{1}{s} + \frac{1}{s'} = \frac{1}{f}, \quad (2.1)$$

that relates the object distance s and the image distance s' to the focal length f [PC82; PP93].

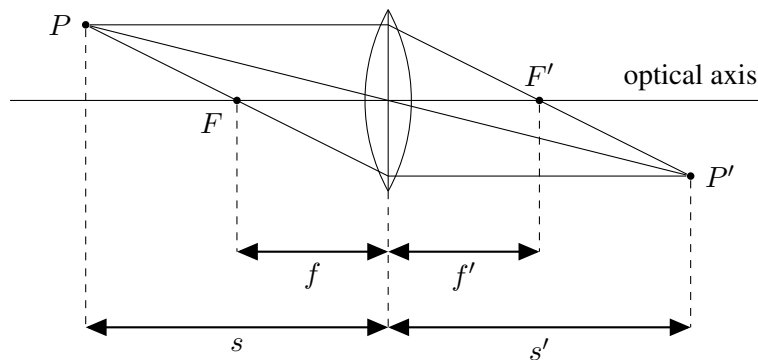


Figure 2.2: Thin lens model, where the lens thickness is omitted such that a ray has only one refraction point.

Focus One notion that is introduced with the thin lens model is focus and depth of field [PC82]. While a pinhole camera displays every object at any distance in focus and therefore has an infinite depth of field, this is not the case with a thin lens. When one wants to focus on an object that lies on a plane with a certain distance s to the lens, the so-called focal plane, the film plane needs to be at a distance s' to the center of the lens. Rearranging Equation 2.1 for s' yields

$$s' = \frac{fs}{s - f}. \quad (2.2)$$

An object behind the focal plane, as shown in Figure 2.3, is projected in front of the film plane, whereas an object in front of the focal plane has its focus behind the film plane. In both cases, the image of the object appears as a blurred circle, rather than a focused point. This circle is called circle of confusion or blur disk [Bar+03].

Thick Lens Model

As opposed to the thin lens approximation, the thick lens model features a finite thickness. Its main use in computer graphics is to approximate more complex lens models consisting of multiple lenses [KMH95; HSS97].

A thick lens model has two principal planes, called primary principal plane H and secondary principal plane H' with a certain distance t to each other. They are defined analogously to the thin lens principal plane, with

2 Background and Related Work

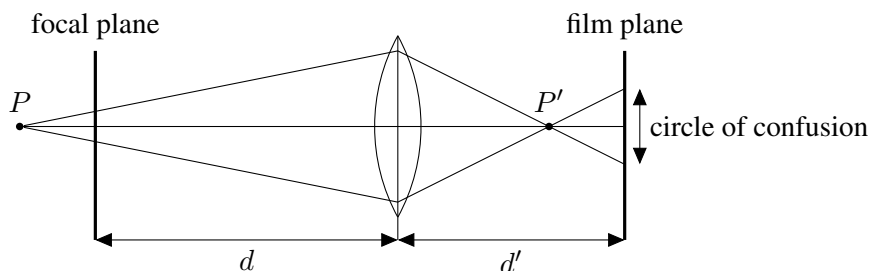


Figure 2.3: The circle of confusion in the thin lens model.

the addition that rays move axis-parallel between the principal planes. To find the position of the secondary principal plane, a ray parallel to the optical axis is refracted by the lens, shown as a dotted line in Figure 2.4, and intersected with the optical axis at F' . Connecting the incident and emanating ray yields a point on the secondary principal plane. In the same fashion, axis-parallel rays from image space meet at the primary focal point F and the intersection of original and refracted ray define the primary principal plane. [KMH95]

The principal planes can either be found by tracing rays through the lens or are calculated analytically using lens thickness formulae [KMH95; Bar+03]. Once they are determined, image generation is calculated in the same way as with the thin lens model, with the only difference being that s now depicts the distance between P and H , while s' is the distance between P' and H' . Following this, focusing an object or calculating its circle of confusion can be derived from the thin lens model, only needing an axis-parallel translation of rays from the primary to the secondary principal plane [Bar+03].

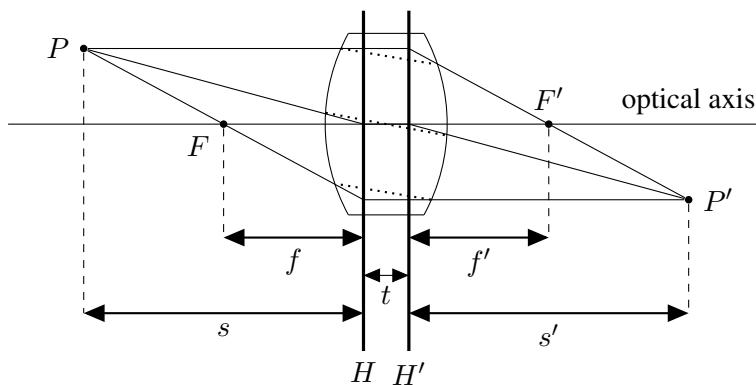


Figure 2.4: Thick lens model featuring two principal planes, between which rays are shifted parallel to the optical axis. Dotted lines show the real paths of light rays.

Realistic Camera

The general consensus of literature is that the previously described models do not allow for accurate simulation of real physics and optical effects [KMH95; Wu+10; SDHL11]. Therefore, models of real photographic lenses have been introduced to computer graphics. Such a lens system consists of a number of lenses and stops centered on the optical axis. Stops are opaque lens elements with circular openings, of which the aperture (or aperture stop or diaphragm) is the stop that limits light rays the most [Bar+03]. Photographic lenses usually have an adjustable aperture to change its size and to therefore adjust the amount of light hitting the film plane and to change the depth of field that increases when the aperture shrinks [KMH95; PP93].

Snell's Law of Refraction Physical accuracy of realistic camera models comes from considering Snell's law of refraction at every surface of the optical system. This formula describes the relationship between the angles

2 Background and Related Work

of incidence and refraction of light and can be applied to any surface that has an intersection point and a surface normal [PP93; Smi00]. In its basic form, Snell's law states

$$n_I \sin \theta_I = n_T \sin \theta_T, \quad (2.3)$$

where n_1 and n_2 are the indices of refraction of the originating and the entering medium, θ_I is the incident angle and θ_T the refraction angle, although other variables are commonly used as well [PP93; Smi00]. For convenience of calculation, the relationship is frequently written in vector form as

$$n_I(\mathbf{I} \times \mathbf{N}) = n_T(\mathbf{T} \times \mathbf{N}), \quad (2.4)$$

where \mathbf{I} and \mathbf{T} are the incoming and outgoing ray directions and \mathbf{N} is the surface normal [Wu+10]. A schematic representation of this relationship is given in Figure 2.5.

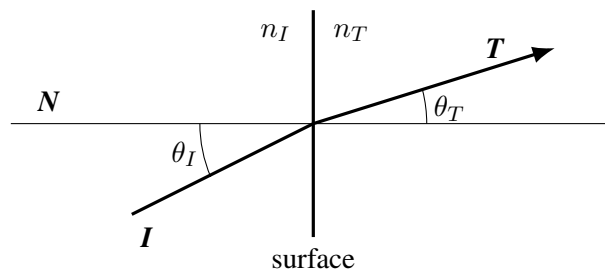


Figure 2.5: Snell's law of refraction relates the angle of incidence and the angle of refraction of a light ray.

Entrance and Exit Pupil When looking through a lens, the image of the aperture stop is called entrance pupil when looking from object space and exit pupil from image space respectively [Nas10; Hec17]. These virtual pupils and the aperture stop are very closely related. If a ray passes through one of them, it passes through the others and therefore through the whole lens system [Wu+10]. The marginal ray of the aperture stop defines the size of the pupils, while the center ray determines the center. Figure 2.6 shows the entrance pupil of a lens consisting of two elements and an aperture stop in between. In such an optimal lens, all points have the same entrance and exit pupil, regardless of their distance to the optical axis [Mah11]. In reality however, many lenses with multiple elements distort the pupils when looking from an angle to modify exposure at the edge of the film [Kin92; Hec17].

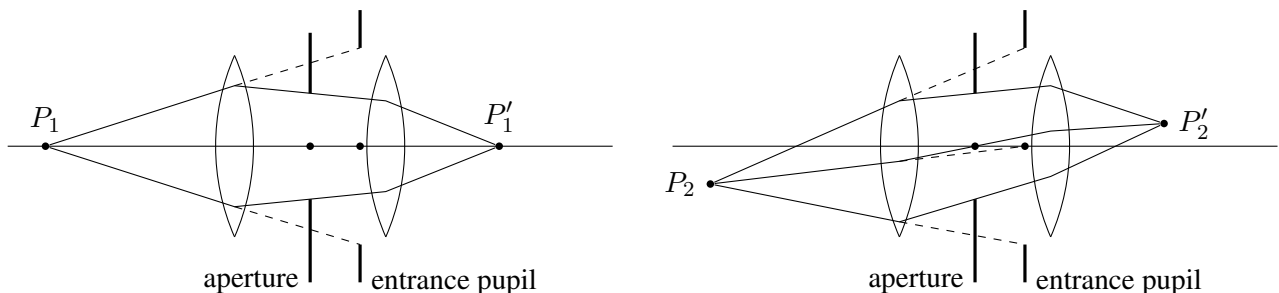


Figure 2.6: The location of the entrance pupil of a lens with two elements for two object points.

Tabular Lens Description Lens systems are typically described in a tabular form, as for example a 50 mm $f/1.8$ double-Gauss lens [Lai95, p.77] in Figure 2.7. While lens manufacturers are cautious when it comes to releasing lens design data, some lens descriptions can be found in patents or collections like the books by Smith [Smi92] or Laikin [Lai95]. Each row corresponds to a lens surface, listed from front (object space) to rear (image space) and provides measurements in millimeters. The first column denotes the signed radius of curvature

2 Background and Related Work

of a surface, none or 0 stands for a planar surface. A positive value indicates convexity when viewed from the object space, while a negative curvature means concavity. The thickness entry measures the distance from the current surface to the next one along the optical axis. Next is usually information about the material between the current and the next surface, either as a material name (e.g. BASF-2) or a combination of the index of refraction at the sodium d line and the V-Number that indicates the change of index of refraction with wavelength. The latter variant is represented in Figure 2.7. The last entry is the diameter of the lens element. [KMH95; WZHL11; WZHX13]

Radius [mm]	Thickness [mm]	n_d	V_d	Diameter [mm]
33.802	5.817	1.6645	35.91	41.66
85.717	0.279			39.12
28.745	6.807	1.6779	55.20	34.54
362.913	2.032	1.6477	33.84	34.54
16.728	9.779			22.61
	10.211			19.05
-15.870	2.057	1.6889	31.14	21.34
142.743	7.899	1.9610	54.81	32.77
-24.277	0.279			32.77
-217.518	5.994	1.6779	55.20	40.13
-37.368	0.279			40.13
77.892	4.597	1.9610	54.81	43.94
-1178.029	35.509			43.94

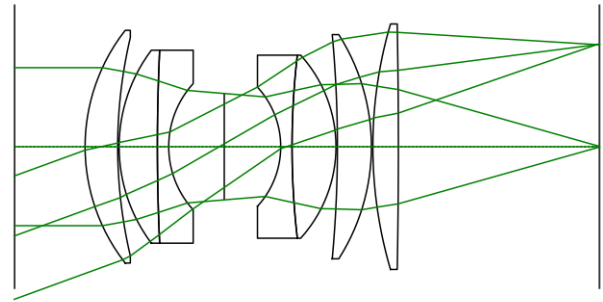


Figure 2.7: Tabular description and profile view of a 50 mm $f/1.8$ double-Gauss lens [Lai95, p.77]. The lens visualization on the right was created using RayOpt³.

2.1.2 Aberrations

Unlike the pinhole camera, a physical camera is not able to create perfect images. Perfect in this sense means projecting a point in object space to a single point in image space. We can describe the point (x', y') where a ray hits the image plane as a function of h and (s, θ) . h denotes the distance from the point (x, y) in object space to the optical axis and (s, θ) are polar coordinates on the exit pupil where the ray intersects it [PP93; Smi00]. Such a projection equation consists of terms of different orders, although only odd-order terms can exist [Smi00; Hec17]. Gaussian optics, considering only first-order terms, was first introduced by Carl Friedrich Gauss [Gau41] and produces optimal imagery as seen in the theoretical camera models previously described [BW99]. On the contrary, the inclusion of higher-order terms can predict departures from perfect imaging, referred to as aberrations [Wel91; PP93; SDHL11]. Monochromatic, aberrations, aberrations that are independent of wavelength, were studied and classified by Ludwig von Seidel and are often referred to as Seidel aberrations [Sei57; Hec17]. Chromatic aberrations occur from varying refraction of different wavelengths [Smi00]. Both types are discussed in the following two segments.

Seidel Aberrations

It has been established that real imaging does not follow Gaussian optics, but departs from perfection. Although higher-order aberrations exist, focus is laid on Seidel aberrations, as they are the most popular and prevalent in reality [Smi00]. There exist five different Seidel aberrations, of which each corresponds to one third-order term and depends on h , s , θ , or a combination of them [Sei57].

Spherical Aberration Spherical aberration denotes the variation of focus with aperture and therefore depends solely on the intersection point radius s [Hec17]. This aberration occurs because different parallel rays have different focal lengths depending on how far from the optical axis they hit the aperture [Smi00]. The further away

³<https://github.com/quartiq/rayopt> (visited on 22nd August 2021)

2 Background and Related Work

from the center a ray hits the lens, the closer its focal point is to the lens. An exaggerated visualization of this effect is shown in Figure 2.8. Depending on where the film plane is located, the resulting circle of confusion can take on different illuminations [Wu+10; Smi00]. Position 1 in Figure 2.8 presents itself as a bright halo with a darker core, while position 2 is a bright core with a fading halo.

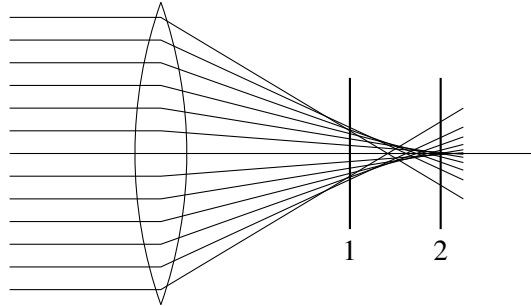


Figure 2.8: Spherical aberration of a simple lens. Different positions of the film plane result in distinct circles of confusion.

Comatic Aberration Comatic aberration, usually abbreviated as coma, depends on all three variables and is therefore a non-symmetric off-axial aberration [PP93]. It denotes the distortion of the image of an off-axis point [Wu+10]. A lens that suffers from coma focuses rays passing through the lens edges on a different height than rays passing through the center [Smi00; PP93]. This results in a comet-like shape on the image plane, as depicted in Figure 2.9. Besides this positive coma, some lenses also exhibit negative coma, where the comet tail points to the image center [PP93; Hec17].

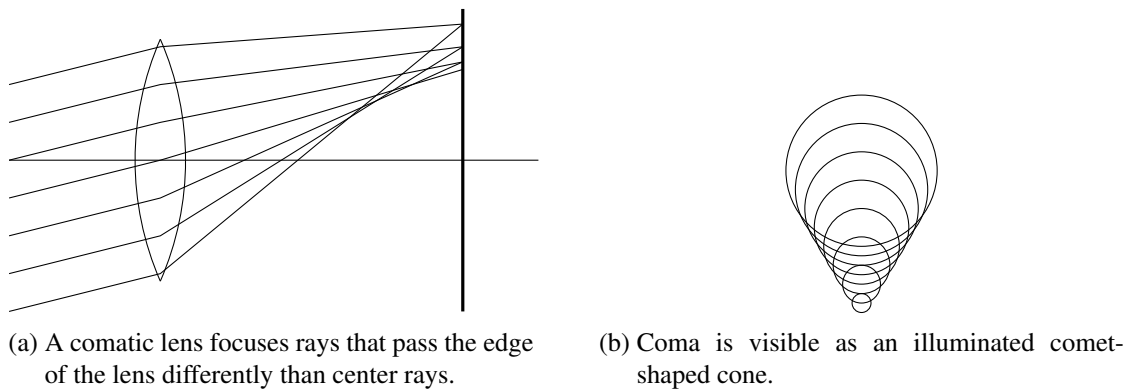


Figure 2.9: Comatic aberration

Field Curvature and Astigmatism Field curvature and astigmatism are related aberrations, as they both depend on the squares of h and s and are therefore often discussed in unison [Wel91; PP93].

Field curvature on one hand changes the focal point with changing angle, turning the image plane into an arc, called the Petzval surface [Wel91; Smi00; Hec17]. Consequentially, the more off-axis an object is, the further away from the image plane its focal point comes to lie and the blurrier its image is. This is depicted in Figure 2.10a.

Astigmatism on the other hand is not symmetrical around the optical axis [PP93]. It occurs because a ray's height varies in two directions [SDHL11]. These two orthogonal directions are called tangential and sagittal planes and are visualized in Figure 2.10b. When astigmatism is present, the tangential and sagittal images do not coincide and one object point is imaged as two focal lines [Smi00; Hec17]. Depending on where the focal point

lies, the point appears as a line stretched in tangential or sagittal direction. When astigmatism is eliminated and tangential and sagittal images match, they result in the Petzval surface [PP93].

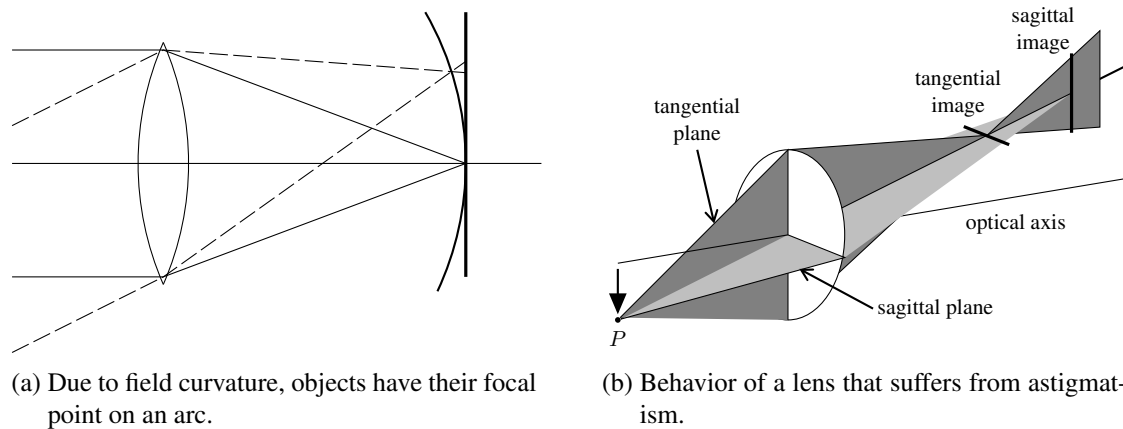


Figure 2.10: Field Curvature and Astigmatism

Distortion The last Seidel aberration is distortion and occurs even if all others have been eliminated and in contrast to the others, does usually not affect sharpness [Hec17]. Instead, straight lines are magnified differently with different distances from the optical axis, visible in Figure 2.11. If the magnification increases with distance from the axis, the effect is called pincushion distortion. The opposite effect is called barrel distortion and stems from decreasing magnification with distance. [PP93; Hec17]

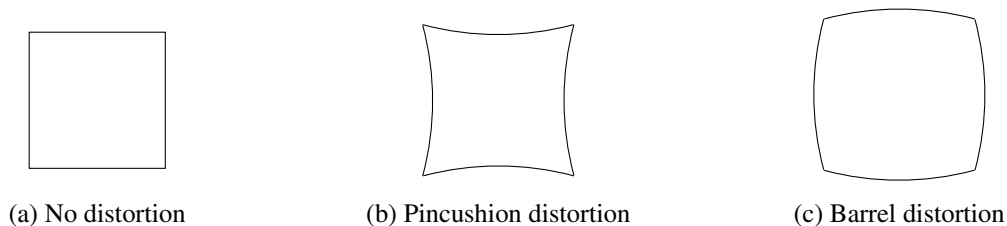


Figure 2.11: Different distortions depending on whether magnification increases or decreases with distance from the center.

Chromatic Aberration

Lens materials not only have different refractive indices, but the index of refraction of a material varies with light wavelengths [Smi00; WZHX13]. This phenomenon is called dispersion and produces another aberration [PP93]. The so-called chromatic aberration comes in two variations, axial (longitudinal) and lateral (transverse) chromatic aberration, displayed in Figure 2.12 [Wel91; WZHX13]. Axial chromatic aberration denotes the effect where different wavelengths have different focal distances, whereas different wavelengths under lateral chromatic aberration are focused at different positions on the focal plane [Wel91]. In photographs, chromatic aberration is most prevalent along hard transitions between dark and light areas and is visible as colored fringes and blurs [WZHX13; Hec17].

2.1.3 Bokeh Effect

The term bokeh refers to the appearance of out-of-focus points [Wu+10]. While sometimes unwanted, there is a visual appeal to it and it can greatly improve realism and depth perception of an image, which is why bokeh often serves as a stylistic device for photographers [WZHX13]. Achieving bokeh effects boils down to a suitable

2 Background and Related Work

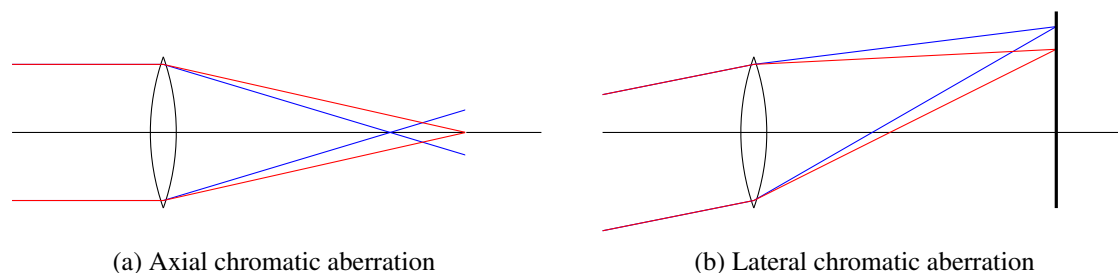


Figure 2.12: Chromatic aberration, where rays of different wavelengths are focused on different points.

combination of aperture, focal length and distance to the objects in and out of focus [Nas10]. Bokeh is closely related to depth of field as both introduce blur to part of the image. Depth of field, however, is only responsible for the amount of blur, while bokeh focuses more on the blur appearance [WZHX13].

The bokeh effect is fairly hard to notice when the out-of-focus areas are of similar brightness, due to many similar circles of confusion overlapping and creating a smooth blur [Nas10]. When the scenery contains specular highlights or distinct light sources, those circles of confusion are prominently highlighted. However, the circle of confusion does not necessarily form a perfect circle. Firstly, when the aperture is stopped greatly (high f-number, small opening), the shape of the bokeh represents the shape created by the mechanical aperture blades [Nas10]. While this shape is normally a polygon, it could theoretically be of a more complex shape by using any desired stencil. Secondly, the location of the light relative to the film influences its shape. Lights at the edges of the film are sometimes not fully circular or regularly polygonal, but appear elliptic. What is among photographers sometimes referred to as cat's eye bokeh or swirly bokeh is caused by the fact that when the aperture is large, off-axis rays might be further restricted by the lens housing, making the exit pupil non-circular [Vor21].

2.1.4 Other Effects

Besides the aforementioned phenomena, there is a variety of additional effects introduced by lens systems, of which the most relevant ones are touched upon here.

Vignetting Another unwanted effect of some lens designs is vignetting. It is closely connected to the exit pupil and refers to the reduction of illumination at the edge of a photograph and is caused by off-axis points having a smaller cone of light that can pass through the lens, resulting in less energy at far off-axis image points [Kin92; Smi00; Hec17]. There are cases where vignetting is wanted as an artistic effect and added retrospectively to an photograph. Stopping down to a small aperture by increasing the f-number reduces vignetting, as a smaller cone of light offers less variability in illumination [Smi00].

Lens Flare Lens flares are unwanted reflections of light in an optical system, especially prominent with few bright lights. A light ray from the scene that enters the lens system might not be refracted by a lens element, but reflected instead. After multiple reflections, the ray can still hit the image plane if enough energy is left, where it leaves a so-called ghost. Because this ray still passes the aperture, the ghost can take on its shape, similar to bokeh, hence why some lens flares manifest as polygonal shapes. Although traditionally unwanted and fought with anti-reflective coatings and lens hoods, lens flares are frequently used in video games or computer-generated movies to convey realism. [HESL11; LE13]

Aperture Diffraction Light rays do not really travel in straight rays, but rather in wave-like motions [Smi00]. Additionally, every point on a light wave front is also a source for secondary wavelets, which can reinforce or interfere with each other and form new wave fronts [PP93; Smi00]. In essence, when a light ray is restricted by

an opening such as an aperture stop, due to diffraction, light can travel in directions other than the original ray's direction [Smi00]. This manifests itself as rings of light of decreasing intensity around an object's image.

2.2 Related Work

Many advances have been made on the topic of realistic camera models in rendering engines, but they can be roughly divided into two approaches: light ray simulation and polynomial approximation. The first solution to a realistic camera is to place a lens system in front of the image plane and to simulate rays passing through it. While tracing rays through all lens elements and calculating the refraction at each element requires additional time, it can be argued that the improved realism far outshines the reduced performance. Nonetheless, different sampling techniques were proposed that aim at keeping this speed overhead contained by making sure that each traced ray makes its way through the lens without getting stuck. More recently, another, more efficiency-focused approach emerged. By treating a lens as a function and using a polynomial to approximate it, tracing rays through the lens becomes needless and performance no longer depends on the amount of lens elements. On the downside, polynomial approximation introduces a certain deviation from the precise solution, and the right balance between performance and error reduction is needed. This fact is especially interesting for real-time rendering, essentially trading accuracy for speed to achieve high frame rates [HHH12; LE13]. Nonetheless, ray-traced samples can be utilized to reduce the approximation error by polynomial fitting, while still preserving the performance.

2.2.1 Ray Simulation Camera Models

In the eighties, the first steps towards a realistic camera model were made by Potmesil and Chakravarty [PC82] and Cook et al. [CPC84]. They extended the pinhole camera by a thin lens model to achieve focus and depth of field in a rendered image.

The foundation to all following approaches was then laid in 1995 by Kolb et al. [KMH95]. They propose a distributed ray tracing algorithm that calculates correct refraction using Snell's law of refraction at each lens element, through which they are able to achieve accurate geometry with all Seidel aberrations as well as correct exposure including vignetting on the film plane. By sampling the exit pupil, which they find through thick lens approximation, the model requires little additional rendering time compared to simpler camera models and is therefore deemed quite practical.

In 2011, Steinert et al. [SDHL11] present a generalization to the model of Kolb et al. for Monte Carlo rendering. They combine spectral Monte Carlo light transport, introduced by Veach [Vea97], with real lens design data and consider physical laws, which is able to produce all aberrations including chromatic aberration. By considering Fresnel interaction at every glass surface, lens flares can be additionally achieved. They further propose a pixel pupil sampling method that stores the passable circle on a pupil plane for each pixel and achieves a passage rate of about 80%.

At about the same time, another model based on Kolb et al. was proposed by Wu et al. [Wu+10] and later improved by the same main authors. Similarly to Steinert et al., they consider real lens data and law of refraction to achieve all Seidel aberrations and propose a sampling method, but do not rely on Monte Carlo for the image generation. As their focus lies on bokeh, they do not consider reflection of light inside the lens and are therefore not able to render lens flare. The second research addresses the shortcomings of the first one, namely chromatic aberration, achieved through a spectral rendering scheme and a dispersive lens model [WZHX13].

The same approach, ray tracing through the lens using the law of refraction, is also used by Pharr et al. [PJH16], who describe a realistic camera model as part of the implementation description of their renderer pbrt. They utilize an optimized version of the pupil computation method of Steinert et al., where exit pupils are calculated for segments along the x axis, which can then be rotated and interpolated to find the exit pupil for any desired point on the film plane. Further, they describe focusing based on thick-lens approximation, where the film plane is moved in z direction by an offset found through a modified thin-lens equation.

2 Background and Related Work

A lens flare focused model was published by Hullin et al. [HESL11]. While this model is technically part of real-time rendering, they argue that omitting certain accelerations would allow for high enough quality to be used in offline rendering, while still being negligibly fast.

Trying to match a real photography lens more accurately, Wu et al. [WZHL11] introduce a model that allows for tunable lens properties. It includes variable focal length and field of view, controllable aperture as well focusing. They further incorporate aspheric lenses, lenses whose profile is not spherical, although only conic sections, such as ellipsoids, paraboloids or hyperboloids are supported [Smi00]. Lastly, a new rendering pipeline allows for accommodation of their new model and integration in the rendering engine LuxCoreRender.

Joo et al.'s model [Joo+16] extends aspheric lens support to all lenses. They provide a ray tracing technique based on numerical root-finding, as such non-conic lenses have no closed-form solution to calculate intersections of rays with surfaces. Further, they incorporate mechanical lens bokeh imperfections that result from the lens fabrication process.

2.2.2 Polynomial Camera Models

A completely different approach to integrating real lenses in rendering engines was first introduced in 2012 by Hullin et al. and coined 'polynomial optics' [HHH12]. They argue that tracing rays through lenses is computationally expensive and either wastes a large part of the ray samples when sampling the outmost lens area or requires sophisticated sampling and filtering methods. To circumvent these problems, they treat a lens system as a function that transforms an incoming ray into an outgoing ray and approximate it using polynomials. By concatenating all lens elements and performing Taylor expansion at the optical axis, their model is able to describe Seidel aberrations with polynomials of third degree or higher, but struggles with wide-angle lenses as the error rises towards the image boundaries. Chromatic aberration is also supported, but requires wavelength as an additional variable in the polynomial. As pointed out by Hullin et al. [HESL11] and later realized by Lee and Eisemann [LE13], first-order approximation of the polynomial allows for real-time lens flare generation.

A first refinement to polynomial optics was presented by Hanika and Dachsbacher [HD14]. Besides implementing their model in a bidirectional Monte Carlo framework, they introduce a fitting procedure, where the polynomial is matched with several thousand ray-traced ground truth samples using least squares optimization. Through this, they are able to reduce the approximation error by 1-2 orders of magnitude compared to Hullin et al. [HHH12]. However, when the field of view is too wide, the error remains too large.

Motivated by the inability to achieve good results with all lenses, Schrade et al. [SHD16] presented yet another variation. Instead of using Taylor series, they obtain a sparse high-degree polynomial with a variant of orthogonal matching pursuit, again by fitting with ray-traced samples. This approach is faster and has a lower error in almost all cases than the previous method.

A further improvement is promised by Zheng and Zheng [ZZ17a], who find the sparse polynomial through an adaptive machine learning algorithm that greedily constructs new terms from existing terms. Further, they introduce light field partition to account for the higher error in off-axis regions. In another paper, Zheng and Zheng [ZZ17b] follow a more data-driven approach and make use of a neural network that learns the mapping between incident and outgoing rays from samples.

3 Model

As elaborated previously, a multitude of physically-based camera models have been proposed, either tracing rays through the lens or approximating the lens using polynomials. While many are not suitable for the scope and objective of this thesis, there are a number of models that need closer inspection and analysis, namely Steinert et al. [SDHL11] and Wu et al. [Wu+10; WZHX13] as candidates of ray simulation models and Schrade et al. [SHD16] and Zheng and Zheng [ZZ17a] as their polynomial counterparts. The following chapter attempts to draw a comparison of them and concludes with a model for implementation. To arrive at this decision, various aspects have to be considered, including render quality, performance, supported effects, supported lens types as well as the given environment appleaseed. A roundup thereof is provided in Table 3.2.

3.1 Model Comparison

Achieving photorealism requires good image quality and very accurate representations of optics and physics. Ray simulation, being as close as possible to physics by using Snell’s law of refraction, naturally achieves flawless image quality. With polynomial optics, this was not always the case, but tended to improve as the years progressed, with the recent models showing no quality losses visible for the naked eye [SHD16; ZZ17a]. Nevertheless, it is important to find the right balance between the polynomial degree and the approximation error, which are indirectly proportional. A too low degree leads to a high and visible error, a too high degree extends the computation complexity and essentially nullifies the performance benefits of the model.

Ray simulation models require additional computations compared to primitive camera models, which can diminish rendering performance to a certain extent. Both Steinert et al. [SDHL11] and Wu et al. [Wu+10; WZHX13], while being very precise about physics and achieving flawless imaging, make no statement about rendering times. One clue is provided by Kolb et al. [KMH95], on whose model they base on, who found that of the total image generation time, ‘10% of that time was spent tracing rays through the lens system’ [KMH95, Sec. 5]. Wu et al. [Wu+10] further point out that because the lens elements are traversed in sequence, heavy computation can be avoided, because the algorithm does not have to find the closest intersection, which reasons the aforementioned relatively small increase in computation.

A main performance factor about ray simulation is the amount of lost rays. When rays are spawned by sampling the outmost lens element, many rays will not make their way out of the lens, as they hit the aperture stop or the lens housing, especially when the f-stop is high and the aperture therefore small. To combat this, the exit pupil can be used to sample only the cone of light that can pass the lens. Wu et al. [Wu+10] propose a two-step algorithm: First, they calculate the marginal rays of this cone through binary search to find the diameter of the exit pupil. Second, the position of the center of the exit pupil is found by ray tracing. While sampling the resulting exit pupil is reliable for points in the center of the film, points on the film’s edge suffer from bad passage rates. The reason is the same as the cause of vignetting, a smaller cone of light from the edges of the film, as the exit pupil is smaller and distorted by other lens elements [Hec17]. As a solution, Steinert et al. [SDHL11] compute what they call pixel pupils, the image of the exit pupil on the last lens element for each pixel on the film. Through this, they achieve about 80% passage rate, although they argue that this number could be optimized to nearly 100% with more precise pixel pupil calculations. A third approach originates from Pharr et al. [PJH16], who do not compute the exit pupil for every film point like Steinert et al. [SDHL11], but rather for points along the x axis that are then interpolated and rotated, which allows for just a fraction of the computation costs. This is possible, as long as lenses are spherically symmetrical. A comparison of passage rates of the different exit pupil

3 Model

calculation approaches is presented in Table 3.1. Steinert et al. [SDHL11] did not use the same exact lens, but a comparable one and do not specify the number of samples per pixel used to find the exit pupil, hence comparison should be done cautiously.

Author	Approach	Ray Passage Rate[%]	
		Double-Gauss Lens 100 mm $f/4$	Fisheye Lens 10.5 mm $f/8$
-	global pupil	36.9	0.5
Wu et al.	single exit pupil	92.2	74.4
Steinert et al.	pixel pupils	-	79.7 ⁴
Pharr et al.	optimized pixel pupils	99.8	88.7

Table 3.1: Comparison of ray passage rates with different sampling methods. The optimized pixel pupils are computed for 50 points at 2000 samples each.

In polynomial optics, evaluating polynomials is 26 times faster in an evaluation by Hanika and Dachsbacher than ray tracing through the lens [HD14]. It is however to note, that Hanika and Dachsbacher use a standalone unit test and a Canon zoom lens with a large number of lens elements for the evaluation. Further, they do not utilize any sampling method for ray generation, meaning that the passage rate of rays for tracing through the lens is very low and yields huge efficiency losses [HD14]. While the evaluation of polynomials is inherently faster and close to thin lens model speeds, the gap to ray simulation is presumably not as large as it might be perceived, especially when looking at the total rendering time. What is costly however, is generating such polynomials in the first place [SHD16]. Only the adaptive algorithm by Zheng and Zheng can build a sparse polynomial in polynomial time, while the orthogonal matching pursuit of Schrade et al. requires an almost exponential number of evaluations [ZZ17a; SHD16]. Unfortunately, there are no concrete comparisons between the two approaches that numerically relate them in a realistic testing environment, the performance aspect is therefore hard to assess.

While the goal of all models is the simulation of a physical camera, achievable effects differ. Steinert et al. [SDHL11] make use of the Monte Carlo light transport simulation with wavelength as additional dimension to achieve all Seidel aberrations, chromatic aberration as well as lens flare and aperture diffraction. Wu et al. [WZHX13] lay focus on bokeh rendering and do not consider reflections of light inside the lens system to create lens flare. Wu et al. [Wu+10] further do not incorporate spectral rendering and can therefore not achieve chromatic aberrations. As for polynomial optics, Seidel aberrations are achieved with polynomials of third degree or higher and chromatic aberration requires wavelength as an additional variable [Sei57; HHH12]. Lens flare rendering can be achieved efficiently using polynomials [HHH12; LE13], but is not directly supported by all models [HD14; SHD16; ZZ17a].

The traditional way of describing lenses as explained in Figure 2.7 only allows for spheric lens elements, limiting all simulation models that iterate through this tabular description. For polynomial optics, however, the insides of a lens are not of importance and treated as a black box, the lens can therefore be of any shape. However, in all viable approximation models, polynomials are fitted from ray-traced samples and are therefore technically limited to the abilities of the ray tracer [HD14; SHD16; ZZ17a].

Lastly, varying suitability for implementing the model in a rendering engine can be observed. While Wu et al. [Wu+10; WZHX13] employ a rather straight forward lens iteration algorithm, Steinert et al. [SDHL11] mainly rely on the Monte Carlo rendering system to do the heavy lifting for them. Schrade et al. [SHD16] provide an extensive library implementation, which could create complications trying to implant it into an existing rendering engine.

⁴Here, a 10 mm $f/8$ Muller Fisheye lens was used instead [SDHL11].

Model	Required for Good Quality	Performance	Sampling Method	Missing Effects	Lens Types
Steinert et al.	-	dependent on lens size	pixel pupils	-	spheric
Wu et al.	-	dependent on lens size	exit pupil	C (2010), LF, D	spheric
Schrade et al.	degree 11 polynomial	fitting: exp eval.: thin lens	-	LF, D	any (spheric)
Zheng and Zheng	30 – 40 terms, no degree limit	fitting: poly eval.: thin lens	-	LF, D	any (spheric)

Table 3.2: Comparison of models for implementation. C stands for chromatic aberration, LF for lens flare and D for diffraction.

3.2 Model Selection

The main decision to be made here is the approach to choose, ray simulation or polynomial approximation, as this is most detrimental for the following implementation. Ray simulation follows real physics, but the additional computation cost reduces the performance. Additionally, global pupil sampling yields bad ray passage rates, requiring precomputation of one or many exit pupils. Its opponent, polynomial optics, offers inherent perfect passage rates and speeds close to the thin lens model, but needs time to build the polynomial and additional consideration about the desired degree of approximation, as the quality directly depends on it.

As appleseed is a physically-based rendering engine, ray simulation is arguably better suited for implementation as it portrays the scene more accurately than the approximation of this simulation. Additionally, the ray simulation model allows for closer implementation to the already existing camera models due to said physically-based nature. Further, the implementation can benefit from already existing classes and methods for steps like sampling, intersections or refractions. Possibly, there is a certain loss of overall performance that needs to be accepted, but as no detailed research was conducted about it, exact numbers remain unknown. Polynomial optics, while having many promising traits, is less suitable for the goal of this thesis to implement a fully parameterizable camera model. A change in focus or focal length changes the whole lens configuration and would require rebuilding the polynomial from scratch. With the ray simulation model, as no large precomputation is necessary, such variable parameters are usable without any drawbacks. Due to these reasonings and the fact that polynomial optics has similar support for optical effects and is in reality also limited to spherical lenses, ray simulation is pursued from now on.

Inside ray simulation, different models provide different characteristics and strengths. The model of Steinert et al. [SDHL11], although featuring many photographic phenomena, was deemed not practical and hard to integrate into Monte Carlo frameworks [HD14]. On top, their sampling method later drew criticism because it requires precomputation and storage for every focus distance and aperture size [SHD16]. An optimized and equally reliable sampling variation is the approach by Pharr et al. [PJH16]. On the opposite, Wu et al. [Wu+10] sample a single exit pupil, which requires virtually no precomputation, but is not as precise as Steinert et al. [SDHL11]. Nonetheless, the sampling approach of Wu et al. [Wu+10] will be pursued in this thesis for two main reasons: First, computation is much more efficient and the ray passage rate is only slightly below the others. Second, by computing only one exit pupil, it tends to be larger than the actual exit pupil, especially in edge regions. This allows for preservation of some lens effects that are erased with more precise exit pupil calculations, such as vignetting. For a similar reason, different bokeh shapes, depending on the film location, appear more naturally and increase realism. Concerning the algorithm to trace a ray through the lens, all ray simulation models utilize the algorithm by Kolb et al. [KMH95], whose approach will be used as well in this thesis.

4 Implementation

The main goal of this thesis is the implementation of a physically-based multi-lens camera model in the open-source renderer `appleseed`. As reasoned previously, a ray simulation model is the more sensible choice within the scope of this thesis as well as from a maintainability point of view. Wu et al. serve as a basis for the implementation of the pupil calculation and the main sequential tracing algorithm [Wu+10] and provide the foundation for tunable lens properties [WZHL11], while Kolb et al. [KMH95] lay the foundation for the tracing algorithm. The complete source code is available on GitHub⁵ and will be merged into `appleseed`'s repository⁶.

4.1 Lens Specification

As a very first step, the lens specification file set by the user is read into the system. It follows the notation of Figure 2.7, but the V-number is dropped, as it would only be relevant when considering dispersion. If the file consists of valid lines of radius, thickness, index of refraction and diameter separated by spaces, the lens is stored as a standard library vector. Next, the lens specification is modified according to the inputs of the user.

Focal Length and Field of View These two properties are inversely proportional and either can be modified to achieve the same result. Most available lenses have a fixed focal length and field of view, the only exception are zoom lenses with moving lens elements to accommodate for the change of focal length. A simpler way of changing a lens prescription to any desired focal length, however, is scaling each lens element by the ratio of the new focal length and the original focal length [Smi92; WZHL11]. A 50 mm lens that is desired to have a focal length of 100 mm therefore needs to be doubled in size. Thick lens approximation based on Pharr et al. [PJH16] is used to determine the focal length of the lens. For that, a horizontal ray is created and traced through the lens. Next, the intersection of the ray with the optical axis yields the focal point, while the intersection of the ray with the original ray determines the location of the principal plane [PJH16]. The difference between the focal point and the principal plane results in the focal length, like previously shown in the discussion about the thick lens model in Section 2.1.1. Lastly, each lens element's radius, thickness and diameter is multiplied by the ratio of the user input focal length and lens focal length.

Aperture Size and Shape A photographer on one hand measures his aperture size as an f-number or f-stop. It is given by the ratio of the focal length and aperture diameter of the lens [Hec17]. A change in f-number can therefore be achieved by modifying the diameter of the aperture while leaving the focal length unchanged. As the focal length is known from the previous step, the size of the aperture can be changed immediately if specified by the user.

On the other hand, the user can modify the shape of the aperture to either reflect the mechanical blades of a real aperture or to achieve artistic effects by specifying an arbitrary aperture shape. This customization feature already exists in `appleseed`'s thin lens camera and is carried over accordingly [Bea+19]. To achieve a polygonal aperture, a number of aperture blades as well as a tilt angle is specifiable. In the other case, an image is set, which serves as a map to build the aperture shape based on the luminance, such as a white star on a black background.

⁵<https://github.com/JaanWilli/appleseed/tree/multilens-camera> (visited on 20th August 2021)

⁶<https://github.com/appleseedhq/appleseed> (visited on 20th August 2021)

4 Implementation

Focus Focusing at a certain point in object space can be achieved by moving the image plane closer to the lens or further away. This needs to be done after changing the focal length and requires a recomputation each time the lens configuration is changed. It is implemented following the approach by Pharr et al. [PJH16] using a modified thin lens formula

$$\begin{aligned}\frac{1}{f} &= \frac{1}{p-z} + \frac{1}{z'-p'} \\ &= \frac{1}{z'-p'} - \frac{1}{z-p},\end{aligned}\tag{4.1}$$

where the object distance s is now the distance from the object plane z to the primary principal plane p and the image distance s' is the distance between the secondary principal plane p' and film plane z' . The principal planes are found as described above. To achieve focus, the two focal points should now come to lie on the film plane z' and the desired object in object space at depth z . The equation therefore needs to be translated by a distance δ , such that

$$\frac{1}{f} = \frac{1}{z'-p'+\delta} - \frac{1}{z-p+\delta}.\tag{4.2}$$

By solving for δ , we arrive at

$$\delta = \frac{-z'+p'-z+p \pm \sqrt{(-z+p+z'-p')(-z'+p'+z-p-4f)}}{2}\tag{4.3}$$

and δ is added to the last lens element's thickness, which defines the distance to the film plane. As δ has two solutions, the smaller of the two is chosen for convenience. [PJH16]

4.2 Pupil Calculation

As established in Chapter 2, the entrance and exit pupil are the images of the aperture and a ray can only pass the aperture if it also passes both pupils. It was further discussed in Chapter 3 that sampling the exit pupil is required, as global sampling would result in inferior passage rates. Wu et al. [Wu+10] compute the exit pupil with a two-step algorithm where first, the marginal ray and second, the center of the exit pupil is found through ray tracing.

The first step is depicted in Algorithm 4.1 and visualized in Figure 4.1. It starts off by defining a minimum and a maximum ray from the center of the film plane to the center and the edge of the last lens element, respectively, denoted as R_{min_d} and R_{max_d} . Then, in a binary search fashion, the marginal ray is approximated iteratively until the cosine similarity

$$\frac{R_{min_d} \cdot R_{max_d}}{\|R_{min_d}\| \|R_{max_d}\|}\tag{4.4}$$

between the minimum and maximum ray direction is close to 1, signifying exact similarity [Wu+10]. For the implementation, 1×10^{-12} is chosen as the deviation threshold, which is reached after about 20 iterations. This way, nearly perfect ray passage rates are achieved with lenses that keep an equal exit pupil for all film points.

Algorithm 4.2 and Figure 4.2 show the second part of the algorithm, in which the center of the exit pupil is found. This part starts off at the center of the aperture stop P_0 . From there, a ray is created towards a paraxial point P_1 on the next lens element. This ray is then traced through the remainder of the lens until it leaves the lens. Lastly, the ray is intersected with the optical axis and the intersection point denotes the center of the exit pupil. For the implementation, P_1 cannot lie on the optical axis as the ray would then be identical to the optical axis and not produce an intersection. Too much distance to the optical axis, however, reduces the accuracy of the intersection point, P_1 is therefore chosen to be 1×10^{-8} of the lens radius away from the optical axis. [Wu+10]

With the center of the exit pupil found, the radius of the pupil can be computed from the marginal ray and is stored together with the center point. The entrance pupil is found following the same schema on the opposite side of the lens, but it is only needed in case a ray should be traced from the scene to the film.

4 Implementation

Algorithm 4.1 Finding the marginal ray

- 1: $R_{min} \leftarrow$ ray from P_0 to P_{min}
 - 2: $R_{max} \leftarrow$ ray from P_0 to P_{max}
 - 3: $R \leftarrow R_{max}$
 - 4: **while** $1 - \text{cos_sim}(R_{min}, R_{max}) \geq \epsilon$ **do**
 - 5: trace R through the lens
 - 6: **if** successful **then**
 - 7: $R_{min} = R$
 - 8: **else**
 - 9: $R_{max} = R$
 - 10: $R = (R_{min} + R_{max})/2$
 - 11: R is marginal ray and determines exit pupil radius
-

Algorithm 4.2 Finding the exit pupil center

- 1: $R \leftarrow$ ray from P_0 to P_1
 - 2: forward trace R through the lens
 - 3: intersect R with optical axis
 - 4: intersection is exit pupil center
-

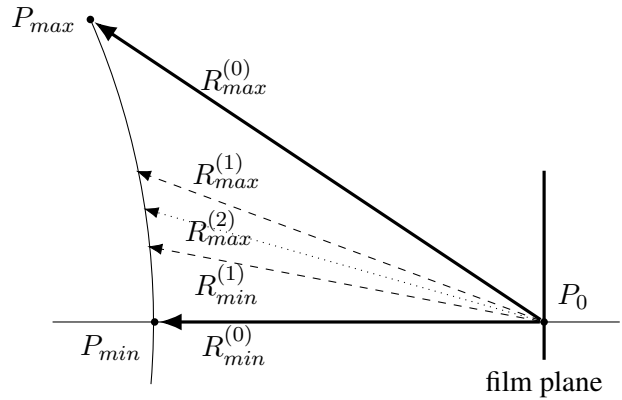


Figure 4.1: Finding the marginal ray from the center of the film plane.

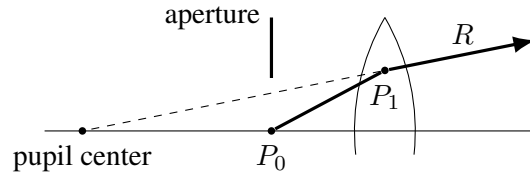


Figure 4.2: Finding the center of the exit pupil.

4.3 Sequential Ray Tracing Through the Lens

At the core of the multi-lens model lies the algorithm to trace a ray through the lens, starting at the film plane, as described in Algorithm 4.3. It was originally proposed by Kolb et al. [KMH95] and is used by all models that rely on ray tracing. In essence, once a new point on the exit pupil is sampled and the starting ray is defined, the container of the lens elements is iterated and for each lens element, the intersection between the ray and the current element is calculated. Appleseed offers already implemented intersection methods, from which the method to intersect a ray with a sphere is used [Bea+19]. As this intersection generally has two solutions, entering and exiting the sphere, the direction of the ray and the curvature of the lens need to be considered to choose the right solution. If a valid intersection exists and lies within the opening of the lens element, the ray is refracted using Snell's law of refraction. At the end of the algorithm, the ray points in a certain direction into the scene and is handed back to the renderer for the actual ray tracing in the scene. If, however, the ray could not be refracted or the intersection lies outside the lens opening, the algorithm is stopped prematurely and the ray is considered dead. The most important steps along the algorithm are described in detail below.

4.3.1 Pupil Sampling

Once the exit pupil is computed, it can be sampled for points to direct rays at. This method is adopted from the existing implementation in the thin lens camera and in the general case, samples a unit disk with a uniform probability density, which is then transformed to a point on the exit pupil [Bea+19]. In the case that a custom aperture shape is desired, the thin lens camera samples the given shape instead of a unit disk, such that all rays are directed at this shape on the lens. As the multi-lens camera does not require this workaround and inherently takes the shape into account through a real aperture, the shape-dependent sampling is not needed and the whole exit pupil is sampled in any case.

Algorithm 4.3 Sequential Ray Tracing Algorithm

```

1:  $R \leftarrow$  ray from sample point on film plane to sample point on exit pupil
2: for each lens element do
3:    $P_0 \leftarrow$  intersect  $R$  with lens element
4:   if  $P_0$  is not inside the element then
5:     end the algorithm
6:    $R_{origin} \leftarrow P_0$ 
7:    $W \leftarrow$  refract  $R$  using the law of refraction
8:   if refraction failed then
9:     end the algorithm
10:   $R_{direction} \leftarrow W$ 
11: return  $R$ 

```

4.3.2 Intersection Validation

Looping through the lens elements, the intersection between the ray and the element is computed first. In the general case, the Pythagoras theorem is used to determine whether the intersection point lies within the element's diameter. For the aperture, however, other calculations are required in case it is not round. As elaborated in Section 4.1, the shape of the aperture can be controlled via a blade number or an aperture map. Internally, this does not affect the lens specification, but it is dynamically considered in the algorithm. If the aperture is polygonal, a ray casting algorithm by Franklin [Fra03] is employed that bases on the Jordan curve theorem. It generates a horizontal ray starting at the intersection point and counts how many times this ray intersects an edge of the polygon. An odd number of intersections signifies that the point lies within the polygon, an even number determines the point to be on the outside. This algorithm works well for many polygons, including concave ones or polygons with holes and only suffers in precision for points very close to or on the polygon's edge, which should not be of vast consequences in this application [Fra03]. For any other custom aperture shape, the same ray casting algorithm could be used as well, but would require finding the coordinates along the edges of the shape. Instead, an importance sampler is utilized, intended for the thin lens camera to selectively sample points, that is built from the aperture map on initialization [Bea+19]. With it, the probability density of the intersection point on the aperture map is found. If it is nonzero, the point lies within the illuminated part of the map and therefore can successfully intersect the aperture.

In both the polygon and the arbitrary shape case, the simple Pythagoras check is done first to omit the subsequent code in case the intersection point is outside the round aperture stop. This certainly improves readability and facilitates debugging, but also positively impacts rendering performance.

4.3.3 Refraction

To use Snell's law of refraction in Algorithm 4.3, it first needs to be transformed into

$$\mathbf{T} = \frac{n_I}{n_T} \mathbf{I} + \left(\frac{n_I}{n_T} \cos \theta_I - \cos \theta_T \right) \mathbf{N}. \quad (4.5)$$

The cosines of the angles of incidence and refraction are then inferred from the law as

$$\cos \theta_I = \mathbf{I} \cdot \mathbf{N}, \quad \cos \theta_T = \sqrt{1 - \left(\frac{n_I}{n_T} \right)^2 (1 - \cos^2 \theta_I)}. \quad (4.6)$$

Combining Equations 4.5 and 4.6 allows for finding the refracted ray \mathbf{T} from the incoming ray \mathbf{I} , the surface normal \mathbf{N} and the ration of indices of refraction n_I/n_T . As this calculation was already present in appleased as part of the vector implementation, it could be used without any change required. [Wu+10; PJH16; Bea+19]

5 Results

A correct implementation of a multi-lens camera model is able to reflect the optical properties of physical cameras discussed in Chapter 2. This chapter aims at showing the capabilities of the implemented model by reproducing said properties using suitable lenses and appropriate scenes. Further, unobtainable effects are named and their unachievability is reasoned.

5.1 Achieved Effects

Many effects and realistic camera properties are made achievable by the implemented multi-lens camera model. Firstly, the model allows for changing lens variables in a way a photographer can do on a real lens and even beyond what the lens is physically capable of. Secondly, many aberrations and effects originating from the imperfection of a multi-lens setup are accurately reproducible. To accomplish this, scenes that isolate certain effects were designed in Blender 2.8⁷ and exported to appleseed using its Blender plugin blenderseed⁸. They are contributed as test scenes to appleseed's testing environment. In other cases, existing test scenes from appleseed's repository could be used. The scenes were then rendered using the multi-lens camera and a suitable lens that suffers from the image defect at hand. The design of all lenses used can be found in Appendix A.

5.1.1 Variable Lens Properties

Usability of the implemented camera model comes from the variety of lens properties that can be changed from the predefined value in the lens description file, namely focal length, f-number and focus distance. Only this variability later allows for comparability, as lenses can be made matching, for example in terms of focal length. Figure 5.1 shows a compilation of the same kangaroo scene rendered with a double-Gauss lens by Tronnier [Tro54] using different lens settings. The signs are between 1 m and 2 m away from the camera with 0.2 m of space between each. In Figure 5.1a, where the aperture is wide open and the frontmost sign is in focus, a degradation of sharpness can be observed further in the back, as well as a vignetting effect. Increasing the f-number and therefore closing the aperture like in Figure 5.1b eases the narrow depth of field and shows all signs in good focus. Increasing the focal length from 35 mm to 55 mm or 75 mm like in Figure 5.1d and 5.1c narrows the field of view and lowers the depth perception. Again, stopping down the aperture like in Figure 5.1e reduces the amount of blur, whereas a focus distance beyond the scene in Figure 5.1f renders all signs in strong blur.

5.1.2 Aberrations

Section 2.1.2 went into detail about aberrations, their physical causes and their consequences. Assuming that the multi-lens camera model accurately models real physics, these aberrations should be visible in some images rendered using this camera model. Unfortunately, in reality, aberrations usually appear in a blended fashion rather than individually. Further, lens manufacturers are constantly trying to combat these imperfections by coming up with new lens designs. By keeping the scenes as simplistic as possible and using older or simpler lenses, which tend to have less aberration corrections, suitable scene and lens combinations can nonetheless be found to showcase the correctness of a given aberration.

⁷<https://www.blender.org> (visited on 5th May 2021)

⁸<https://github.com/appleseedhq/blenderseed> (visited on 16th July 2021)

5 Results

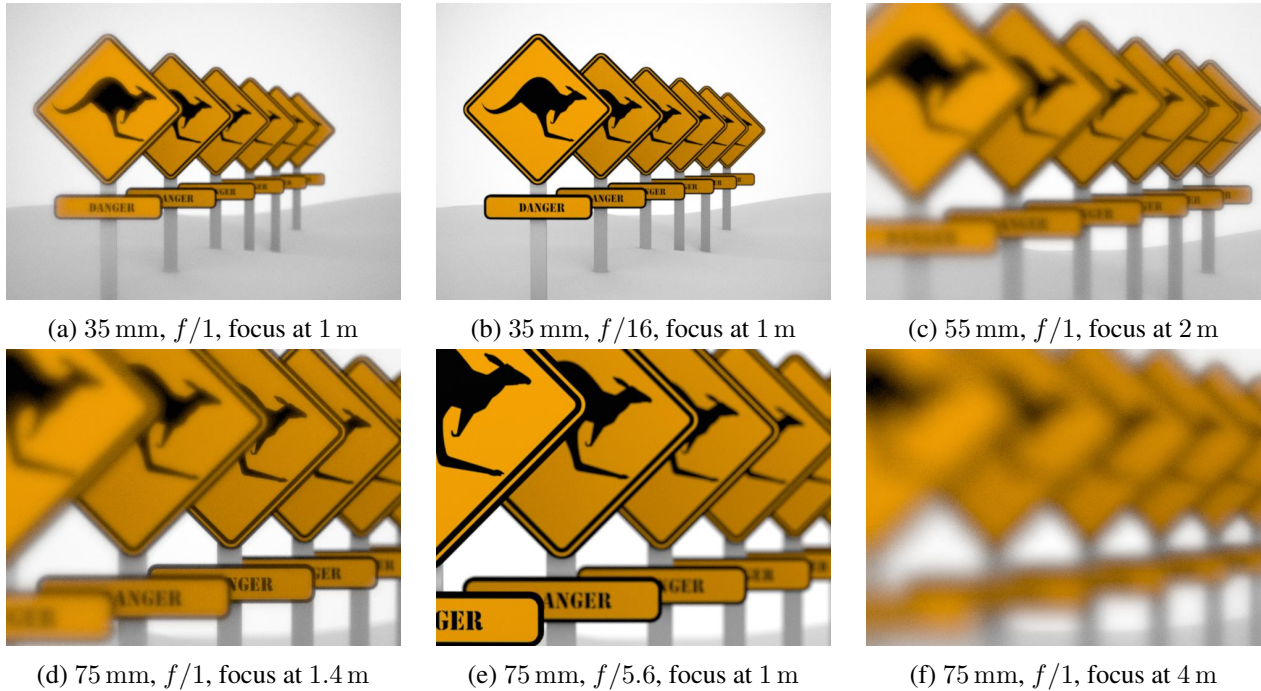


Figure 5.1: The kangaroo scene rendered with the same double-Gauss lens, but set to different variables. The signs in the scene are equally spaced, with the first being 1 m and the last 2 m away from the camera.

Spherical Aberration Spherical aberration describes the effect where parallel object rays do not focus in one focal point and different circles of confusion are created when focusing at different distances. Figure 5.2 shows the same small light source at 1 m distance rendered with the Tronnier double-Gauss lens [Tro54] using five different focus distances, 0.8 m, 0.9 m, 1 m, 1.2 m and 1.4 m. Focusing closer than 1 m results in a dark core with bright halo. Focus distances slightly beyond 1 m turn the light into a bright center with a dark ring around, then at more distance, a dark core starts to appear in the center. It is further noticeable that focus distances further away from the actual distance result in larger circles of confusion that are less bright, as the light intensity is distributed among a larger area.

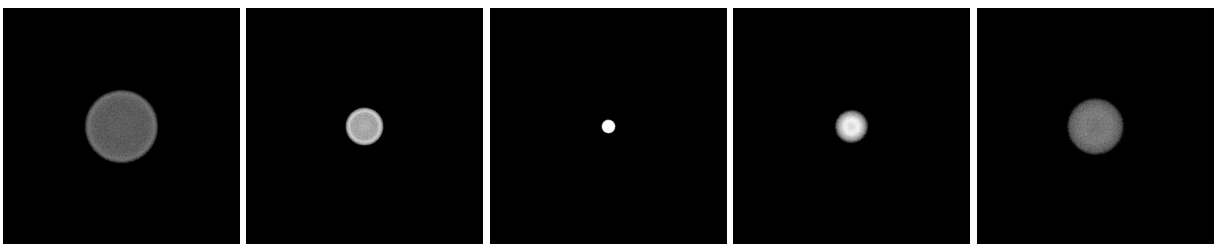


Figure 5.2: Circles of confusion of a light source at 1m distance as a result of spherical aberration. Rendered using a double-Gauss lens at 200 mm, $f/1$, and focal distances of, from left to right, 0.8 m, 0.9 m, 1 m, 1.2 m and 1.4 m.

Comatic Aberration Figure 5.3 shows light spots rendered using a double-Gauss lens [Lai95, p. 75] and a wide angle lens [Ito80] that suffer from coma. The double-Gauss lens has been modified to feature a larger diameter, such that the effect is more prominent. The renders are cropped to display the top right corner, the light on the bottom left therefore lies in the center of the image. Both renders show the coma-typical comet distortion, although the comets of Figure 5.3a are further affected by the extreme exit pupil distortion that stems from the increase in diameter of the lens. It features negative coma, comet tails towards the center, whereas Figure 5.3b

5 Results

has positive coma that distorts outwards. The magnitude of the comet shape is determined by the distance of the light spot to the center of the image, where no distortion is present.

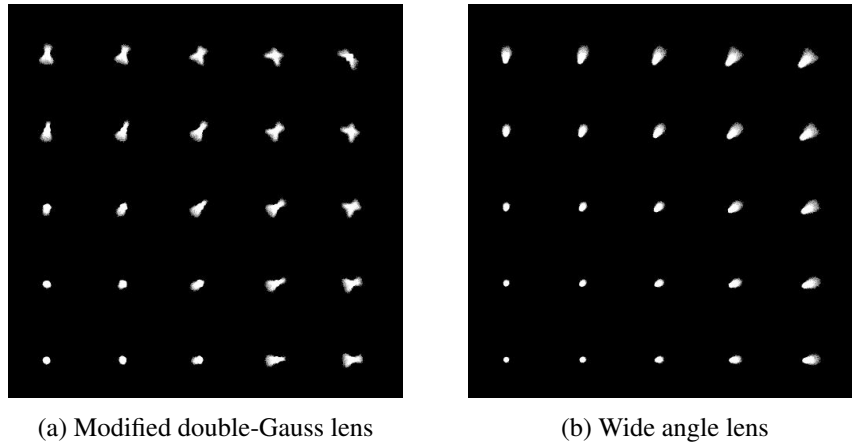


Figure 5.3: Coma distorts small light sources either towards the center of the image (bottom left) or outwards.

Astigmatism Astigmatism can occur in two forms, tangential astigmatism or sagittal astigmatism, depending on whether the tangential or sagittal image is focused on. In tangential astigmatism, when the film plane is located at the sagittal image, a distortion happens in the direction of the optical axis, or in other words, in rays from the image center. Placing the film plane at the tangential image results in sagittal astigmatism and manifests itself as distortion perpendicular to rays from the image center. A suitable way to show astigmatism, is therefore the use of a cartwheel scene, as the spokes and the rims are optimal to prove sagittal and tangential astigmatism, respectively. Figure 5.4 shows such a scene rendered using a Canon lens [Oga96] without astigmatism as reference and a widened double-Gauss lens [Lai95, p. 75] with astigmatism. As only the circular rims are blurred in Figure 5.4b, while the spokes are almost perfectly sharp, this lens suffers from tangential astigmatism.

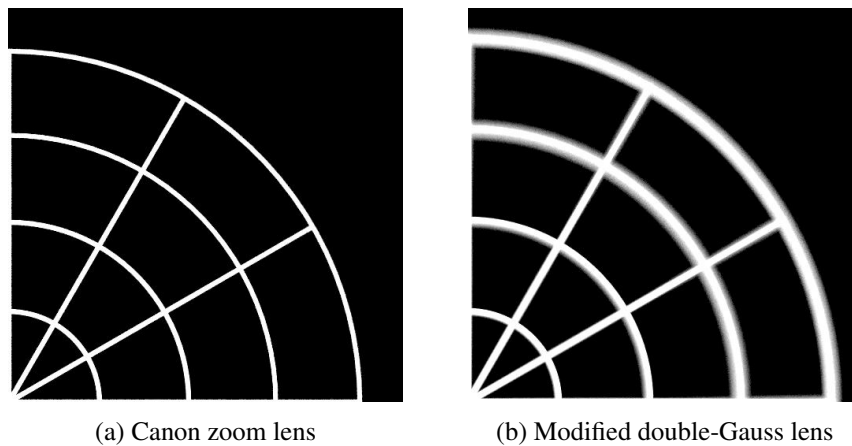


Figure 5.4: Tangential astigmatism blurs only the rims of the cartwheel. The effect diminishes towards the center of the image on the bottom left corner.

Field Curvature Field curvature is present in every lens, but most, if not all, lenses virtually eliminate it. By increasing the diameter of the lens like in Figure 5.5a or by resorting to simple single-element lenses like in Figure 5.5b, the effect can nonetheless be shown. As the image plane is a slight arc, blur starts to appear towards the edges of the image, even though the chess board is in perfect focus. Figure 5.5c, where the focal length is increased, shows less field curvature, as the range of ray angles is more narrow and the extent of the Petzval surface is more limited.

5 Results

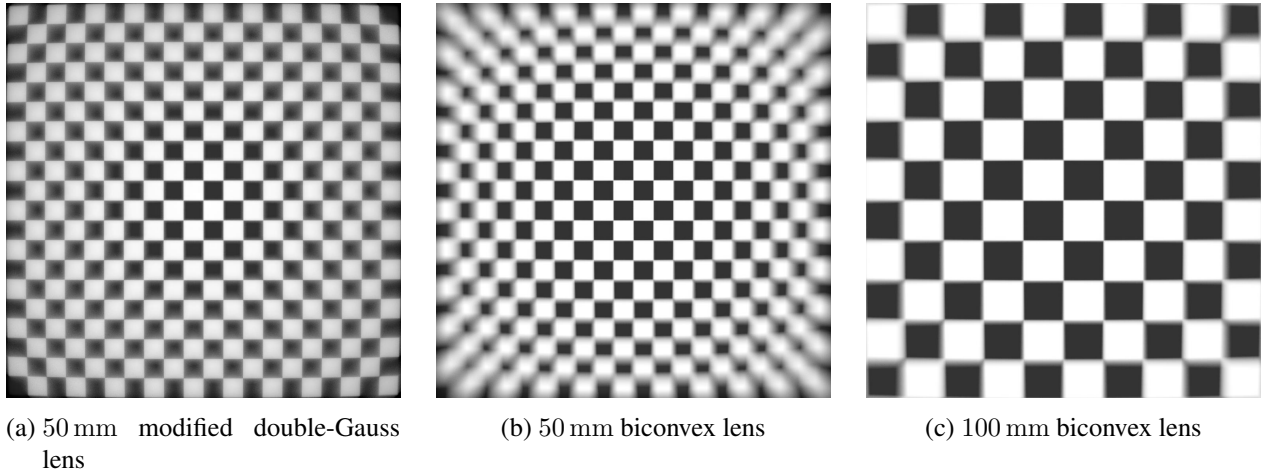


Figure 5.5: The chessboard scene shows the blur caused by field curvature of different lenses.

Distortion The last of the five Seidel aberrations is the effect, that straight lines in the scene are portrayed as curves in the image. All renders in Figure 5.6 were created with equal focal length at a high f-number to combat other aberrations and get sharp lines. Figure 5.6a shows a scene of a grid of equidistant white lines rendered using a pinhole camera. In Figure 5.6b, a biconvex lens [SHD16] creates a pincushion distortion, changing lines to convex arcs, while in Figure 5.6c, a fisheye lens [Miz07] produces a concave barrel distortion.

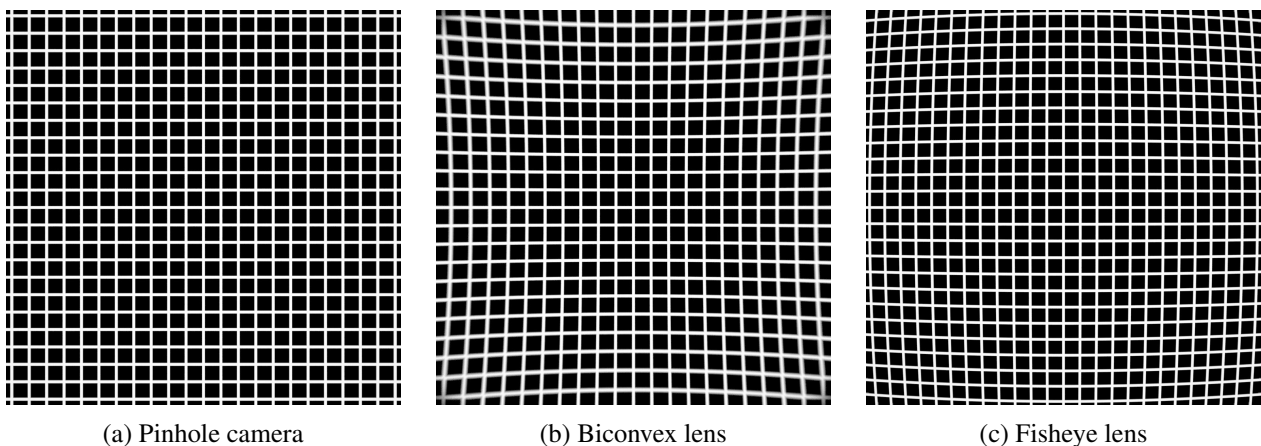


Figure 5.6: A grid of lines rendered with different lenses at the same focal length. While the pinhole camera does not distort the image, the others create a pincushion and barrel distortion, respectively.

5.1.3 Vignetting

The Tessar lens by Brendel [Bre58] introduces strong vignetting at low f-numbers, as visible in Figure 5.7. At $f/1$ for instance, a dark taint can be observed in the periphery of the image. At $f/4$, the effect is reduced and at $f/8$, only the corners are slightly tainted. This is in line with the optical explanation of vignetting, where a large aperture leaves more room for rays of off-axis points to be limited by the rims of lens elements, whereas a small aperture leads to a more consistent exit pupil size regardless of the film position. The fact that vignette is still present even at very small apertures is attributable to this specific lens being at its limit of feasible focal lengths. If the focal length was any smaller and the film plane moved closer to the camera, then off-axis areas would not be reachable by any light and stay completely black. This cannot be prevented by increasing the f-number, as it only makes the transition less fluid, but would require an increased focal length.

5 Results

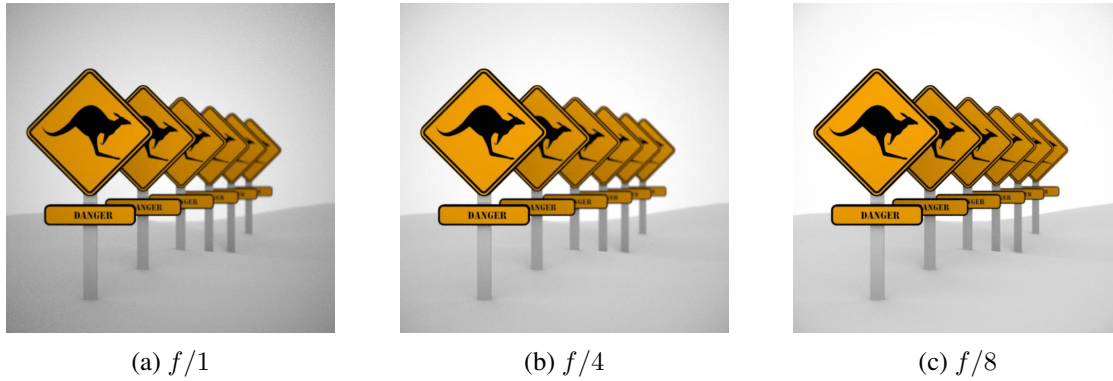


Figure 5.7: Influence of the f-number on vignetting.

5.2 Unachieved Effects

There are certain imperfections and effects that cannot be simulated with the implemented model, namely chromatic aberration, lens flare and aperture diffraction. Although their incorporation is not impossible, they lie outside the scope of this implementation due to high implementation complexity and increased computational needs.

Chromatic aberration requires light refraction based on wavelengths. For the implementation, each ray therefore needs a certain wavelength associated, that is then considered in refraction computation. As the architecture of appleseed merely expects the camera models to spawn rays based on pixel inputs, making wavelength accessible inside the camera would require major refactorings with large impacts on many classes, and is therefore not supported by the current model.

Further, neither reflections, nor ray direction changes are supported inside the lens, both of which would be needed to render lens flare. As one ray can be reflected multiple times and branch into multiple lower intensity rays, a straightforward implementation is impossible for performance reasons and a more sophisticated algorithm is required, which deviates from the scope of this thesis.

Lastly, aperture diffraction was similarly deemed out of scope and of too low impact for the high complexity, which is why it is ignored in the model.

6 Analysis

In this section, the implementation of the multi-lens camera in appleseed is analyzed in detail and compared to the commonly used implementations of the pinhole and thin lens camera. Firstly, a comparison of features is drawn and image quality and realism, which base on those features, are explained and interpreted. In a following part, the performance of the implementation, the main handicap of the multi-lens camera, is closely inspected, both in relation to other camera models and different lens complexities.

6.1 Feature Range

By presenting achievable and unachievable effects in Chapter 5, an implicit range of features of the multi-lens implementation has already been established. A condensed overview thereof, together with an overview over other camera models is presented in Table 6.1. While the pinhole camera is limited to setting a focal length to adjust the image section, the thin lens camera offers depth of field and selective focus through an aperture that is adjustable in both size and shape. The multi-lens camera, as proven previously, additionally has third-order aberrations, spherical aberration, coma, astigmatism, field curvature and distortion, as well as vignetting in its repertory, but is unable to produce chromatic aberration, lens flare and aperture diffraction. Besides a spherical and an orthographic camera, which have no relevancy in this thesis, appleseed lastly offers a fisheye camera that can create barrel distortions in a similar fashion as the multi-lens camera. There, the user can choose different types of projection, which offers versatility, but as its underlying model is the pinhole camera and it merely distorts ray directions, its feature range is equally narrow as the pinhole camera and therefore not comparable to the multi-lens camera.

While the pinhole and thin lens camera are consequentially very limited in effects beyond depth of field and fall behind the multi-lens camera, appleseed offers various effects as post-processing stages, most importantly for this thesis chromatic aberration and vignetting. While the multi-lens camera cannot produce chromatic aberration and therefore has to rely on this post-process, it comes with natural vignetting, which has the potential to be much more nuanced and lens-specific than post-processing vignette.

6.2 Realism

Appleseed's thin lens camera model has been established to be capable of producing depth of field, which can be argued to yield a basic bokeh effect when the focus lies much closer than the object. However, because realistic bokeh is closely related to the optical aberrations, a multi-lens camera model is required to achieve photorealistic bokeh effects. Figure 6.1 shows a scene of many small light sources at different distances to the camera, which lets the dots appear in different sizes. In both renders, the focus distance is set to a small number much closer to the camera than the lights are, such that they appear out of focus. Figure 6.1a was rendered with the thin lens camera, Figure 6.1b with a double-Gauss lens [Tro54] that was tweaked until the bokeh sizes matched, as different lenses yield different bokeh intensities. Looking at the thin lens render, blurred circles of light can be observed, where the brightness is dependent on the proximity of the light source. Comparing it to the double-Gauss render, the used lens introduces bokeh variation and optical aberrations that influence the perception of the bokeh. There, the bokeh is no longer perfectly circular and of equal size, but takes on various shapes and sizes. This is due to the exit pupil neither being circular nor of equal size for all film points. Shape-wise, center bokeh is comparable to the thin lens, as only the aperture determines the cone of passable light. Lights far off-center, however, appear

6 Analysis

	Pinhole Camera	Thin Lens Camera	Fisheye Camera	Multi-Lens Camera
Focal Length	✓	✓	✓	✓
Depth of Field	✗	✓	✗	✓
Focus	✗	✓	✗	✓
Aperture Size and Shape	✗	✓	✗	✓
Spherical Aberration	✗	✗	✗	✓
Coma	✗	✗	✗	✓
Astigmatism	✗	✗	✗	✓
Field Curvature	✗	✗	✗	✓
Distortion	✗	✗	✓	✓
Chromatic Aberration	✗*	✗*	✗*	✗*
Vignetting	✗*	✗*	✗*	✓
Lens Flare	✗	✗	✗	✗
Aperture Diffraction	✗	✗	✗	✗

* Available as a post-processing stage.

Table 6.1: Overview of available properties and achievable effects of camera models in appleseed.

elliptical due to the lens housing being an additional ray-limiting factor. On top, aberrations impact the bokeh further, in this render especially spherical aberration. It manifests itself as bright rings around the bokeh, best visible in the center. In peripheral areas, the effect is diminished, because the region where light hits the aperture is smaller.

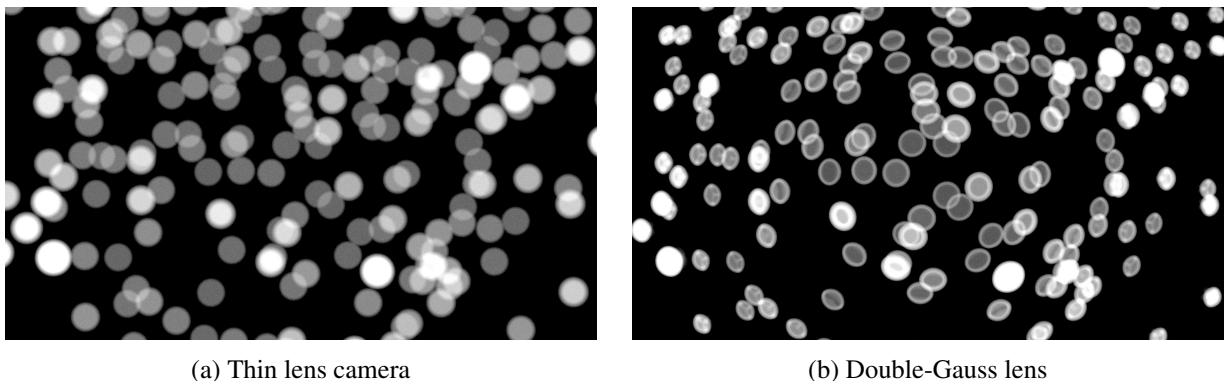


Figure 6.1: Comparison of bokeh effects in a scene of different sized out of focus light sources.

The thin lens model in appleseed further allows for specification of a custom aperture shape, in which lens points are then sampled. On the contrast, the implemented multi-lens model consists of a real aperture that can be modified in shape and size, the workaround of sampling a specific area is therefore not necessary. To show the differences, Figure 6.2 opposes the rendering results of different aperture shapes of the thin lens camera and a double-Gauss lens [Tro54]. The thin lens camera in Figure 6.2a and 6.2c changes the circular blur spots into other shapes, which certainly gives an artistic look, but cannot convey photographic realism. To achieve it, variation of the bokeh depending on the relative location is required. Looking at Figure 6.2b, one can see perfectly hexagonal shapes in the center, that further outwards lose their shape due to no longer being restricted by only the aperture, but by other lens elements. Similar to Figure 6.1b, optical aberrations are visible as well. Figure 6.2d shows the same behavior using a star-shaped aperture, but to a lower extent, as the star pattern reduces the bokeh effect twofold. On one hand, the star being smaller than the hexagon acts like a higher f-number and reduces bokeh. On the other hand, spherical aberration and a change in shape is less noticeable, as it mostly impacts the edges of the bokeh shape. On closer inspection, distorted stars with shorter rays are observable in the corners of the image nonetheless. In both multi-lens renders, a reduction in brightness compared to their thin lens counterparts can be

observed. This is attributed to the implementation, where many rays end at the aperture due to its shape, and hence cannot contribute to the illumination calculation. In the thin lens case, the rays are compacted to a certain shape and the same number of rays as with a circular aperture reaches the scene, achieving higher brightness.

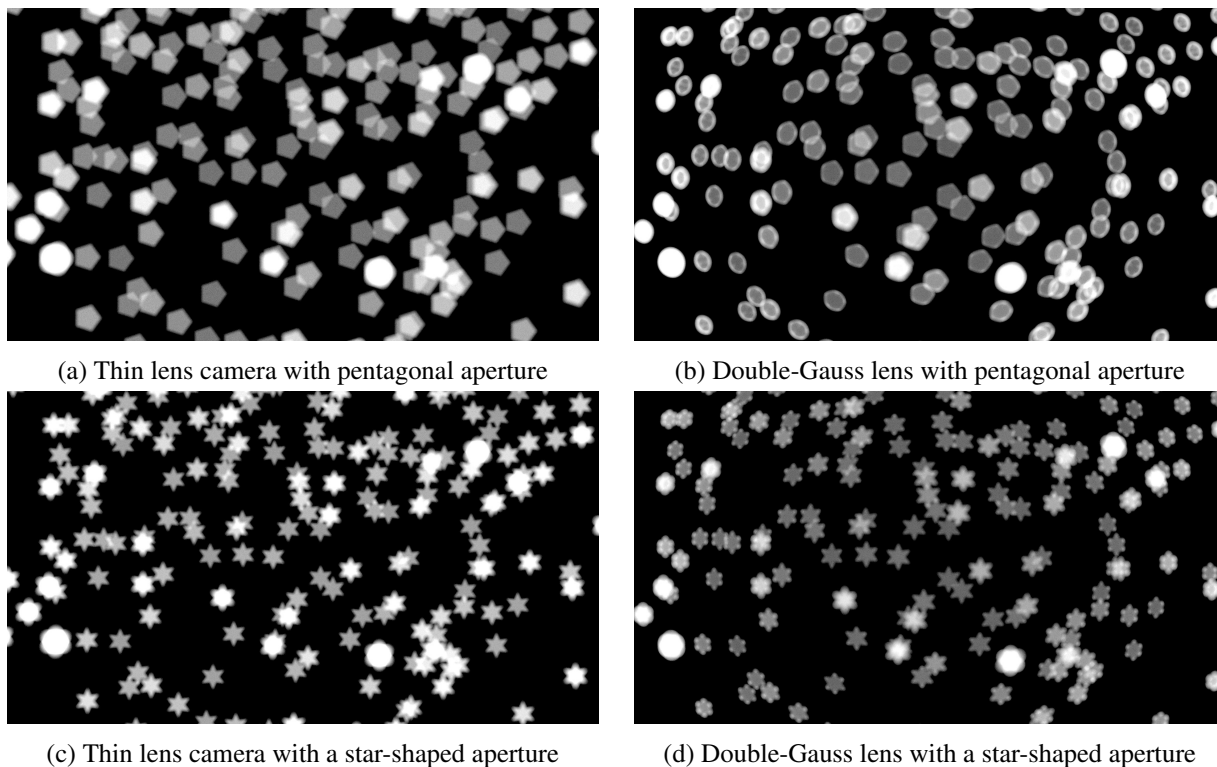
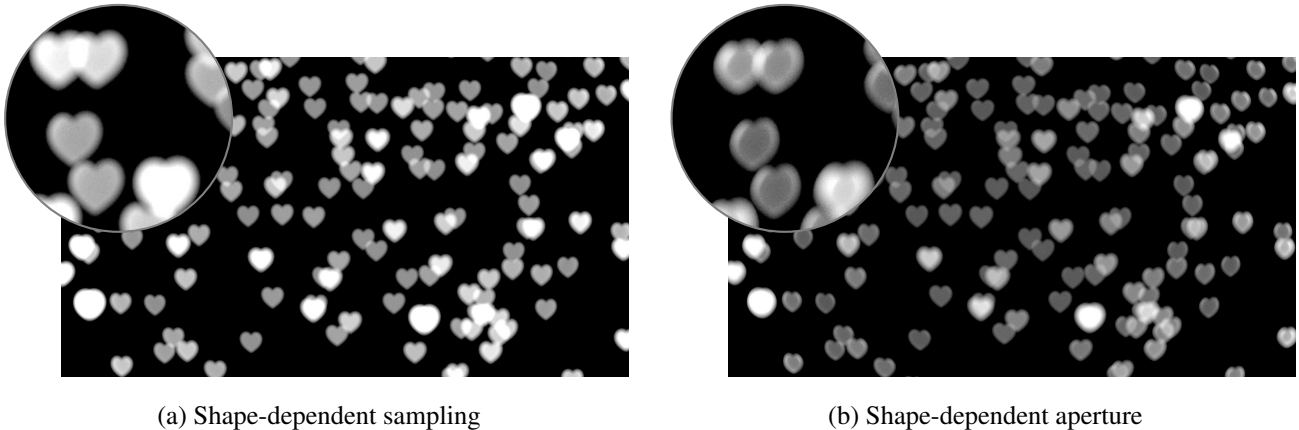


Figure 6.2: Comparison of bokeh shapes between the thin lens and multi-lens camera in a scene of different sized light sources.

For further comparison of the two approaches to achieve custom aperture shapes, both were implemented in the multi-lens camera and are compared in Figure 6.3. Figure 6.3a contains the result of shape-dependent sampling following the thin lens approach, whereas Figure 6.3b was rendered with the real aperture shape consideration described in Section 4.3.2. Both images were rendered using a heart-shaped aperture on a double-Gauss lens [Tro54]. Comparing the two, the afore-reasoned difference in brightness is visible again. Further, looking at the magnified corner section, certain differences can be observed. Firstly, shape-dependent sampling yields a more uniformly bright bokeh, whereas the shape-dependent aperture produces spherical aberration and illuminates the bokeh outlines. Secondly, considering the real shape while ray tracing results in more shape variation than the sampling approach, visible as slightly squeezed and distorted hearts in Figure 6.3b.

Besides bokeh as a visualization of a variety of imperfections that lack in other camera models, vignetting is an additional effect that previous models cannot inherently achieve. As vignetting can be a stylistic device, *appleseed* offers it as a post-processing stage and allows for easy adjustment of the vignette intensity and the anisotropy, the degree of deviation from perfectly circular vignetting. To show the differences of this post-process to real lens vignetting, white squares were rendered and are shown in Figure 6.4. Even though the post-processing vignette intensity was tried to match the vignette of the two real lenses closely, certain differences, mainly concerning the light falloff rate, can be observed. On one hand, the Tessar lens [Bre58] in Figure 6.4a has a smaller bright center and darker edges than its post-process counterpart in Figure 6.4b. On the other hand, while the double-Gauss lens by Hudson [Hud68] in Figure 6.4c and the post-process in Figure 6.4d have similarly bright cores, brightness falls off slower in the double-Gauss image, but reaches a higher darkness in the corners. Post-processing vignette, indifferent of intensity, always features equal falloff rates, visible when comparing Figures 6.4b and 6.4d. Different real lenses, in contrast, have different light falloff at the edges, also depending on focal length and f-number. Consequentially, the multi-lens camera offers advantages in terms of vignette quality by

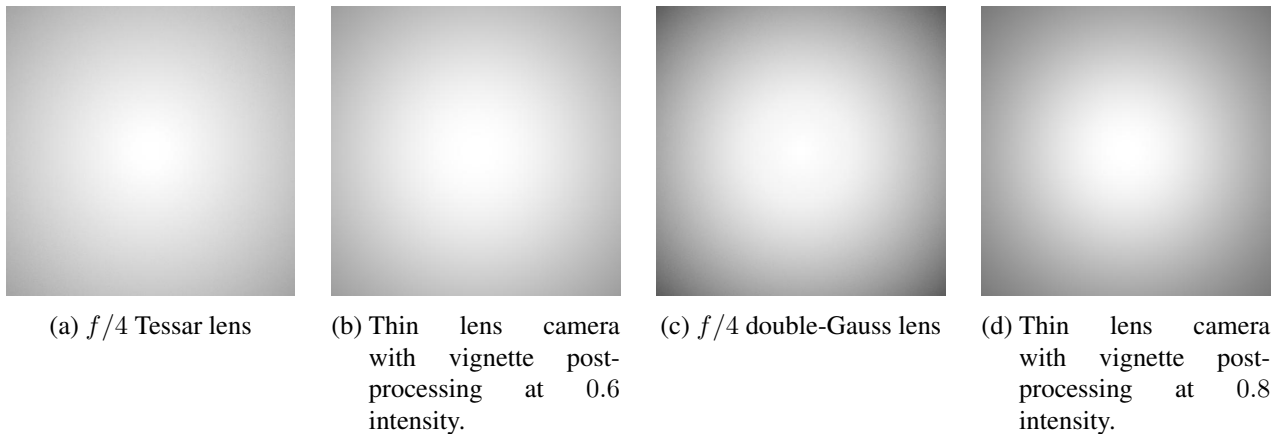


(a) Shape-dependent sampling

(b) Shape-dependent aperture

Figure 6.3: Comparison of two implementations to achieve custom bokeh shapes. On the left, the exit pupil is sampled in a heart shape like the thin lens implementation does it. On the right, the shape is considered when intersecting rays with the aperture.

providing a more realistic and diverse effect. While the post-process offers more customizability, the multi-lens camera ties a distinct vignette to a specific lens at a certain setting. On top, handling post-processing becomes superfluous with the new model.

(a) $f/4$ Tessar lens

(b) Thin lens camera with vignette post-processing at 0.6 intensity.

(c) $f/4$ double-Gauss lens

(d) Thin lens camera with vignette post-processing at 0.8 intensity.

Figure 6.4: Comparison of native vignetting and vignette added through a post-processing step.

6.3 Performance

The camera's performance is a critical measure and can be a deciding factor for appeal and the willingness to use it. Table 6.2 therefore compares render times of different scenes, camera models and lenses. All scenes stem from appleseed's test scene repository [Bea+19]. They were rendered in appleseed studio on an 8-core Intel Core i7-9700 at 3 GHz and 16 GB DDR4-RAM at 2400 MHz with one pass at 100 samples per pixel. Besides appleseed's already existing pinhole and thin lens camera, the following lenses were used in the multi-lens camera: A lens consisting of only an aperture as a reference point, a single biconvex lens [Inc21], a Tessar lens by Brendel [Bre58], two double-Gauss lenses by Tronnier [Tro54] and Angenieux [Ang55], two wide angle lenses by Itoh [Ito80] and Ikemori [Ike82] and a Canon zoom lens by Ogawa [Oga96]. The resolution of each render is 512×512 , the focal length 50 mm and, where applicable, $f/4$ is chosen as aperture size. The numbers for each scene represent the average render time in seconds over three runs. The individual times as well as all rendering output images can be found in Appendices B and C. For the fluffy cornell box scene, a BVH tree is built on the first render after loading the scene, which was excluded from the measurements. Although all scenes

6 Analysis

were rendered with the same settings, great time differences can be observed, which attribute to the fact that the fluffy cornell box scene, for example, requires much more ray computations in the scene than the kangaroo scene does. In the flower scene, while the flower itself requires heavier computation, its black surroundings do not and are processed comparatively much faster. Not only in terms of the scene do render times differ, the choice of camera is also influential. The pinhole camera, being the simplest model, is fastest overall, closely followed by the thin lens camera. The multi-lens camera is slowest, as it requires additional time for tracing each ray through the lens. N_r signifies the number of refractions, intersections with lens elements other than the aperture. Aperture with zero refractions serves as a base case, as it does not feature any lens piece, but solely an aperture. To achieve similar sharpness as other renders and prevent rays in additional directions, the aperture diameter was set at a very small value, such that it resembles the pinhole camera in design, but uses the multi-lens implementation for ray generation. Looking at the speed of this base case in the three scenes, render times very similar to the pinhole and thin lens camera can be observed, which shows that precomputations as well as recurring operations, such as exit pupil sampling and aperture intersection computation of the multi-lens camera are comparable in complexity to the pinhole and thin lens camera. Thus, the render time of the multi-lens camera and therefore the overhead over the other models is almost solely dependent on the number of lens elements the lens consists of, or more precisely, the number of refractions that need to be computed.


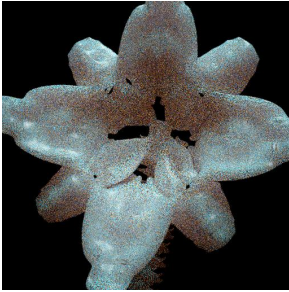
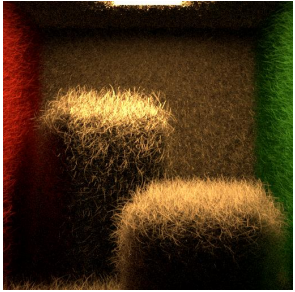
			Kangaroo Scene	Flower Scene	Fluffy Box Scene
					
Camera	Lens	N_r	Render Time [s]	Render Time [s]	Render Time [s]
Pinhole	-	-	5.90	17.10	110.17
Thin Lens	Thin Lens	1	6.37	17.30	110.84
Multi-Lens	Aperture	0	6.40	17.35	110.61
	Biconvex	2	7.75	20.31	116.38
	Tessar	7	10.09	22.03	110.80
	Tronnier	10	12.34	24.84	116.49
	Angenieux	14	14.56	26.01	115.82
	Itoh	18	16.85	28.47	120.81
	Ikemori	25	21.09	31.17	127.01
	Canon	33	28.15	37.75	133.19

Table 6.2: Average render times over three runs of different scenes at 100 samples per pixel using the pinhole camera, thin lens camera and multi-lens camera. The focal length is always 50 mm, the scenes are focused and the thin lens and multi-lens camera are set to $f/4$. The multi-lens model is tested using an assortment of different lenses of different complexity, indicated by their number of refractions N_r .

6.3.1 Multi-Lens Overhead

Figure 6.5 plots the time difference to the pinhole camera for each multi-lens camera lens, indicated by its number of refractions, and each scene. Overall, a rough linearity and the aforementioned strong correlation between refractions and time can be observed. To verify the linearity, Figure 6.6 visualizes the time overhead divided by the number of refractions. For the most part, a constant overhead of about 0.6 s per refraction can be observed,

6 Analysis

validating that the time overhead is almost purely based on the number of refractions. Apart from that, two outliers are visible in the two illustrations.

First, the Tessar lens (7 refractions) and the Angenieux lens (14 refractions) are exceptionally fast, especially in the fluffy cornell box scene. This behavior can be attributed to the fact that both lenses suffer from vignetting, best visible in the kangaroo scene renders in Appendix C. Towards the corners of the image, an increasing percentage of rays are stopped somewhere inside the lens, which reduces the overall number of refractions that need to be computed and in turn benefits the render time. As the illumination computation complexity per ray is highest in the fluffy cornell box, fewer rays due to vignetting reduce the render time the most. In the other two scenes, corner rays are shot into the nowhere anyways, vignetting therefore has only a minor impact. By modifying the f-number and reducing vignetting, one could diminish this time deviation, however changing the f-number can impact the overall performance. Concretely, a large aperture has a worse ray passage rate for the same reason that is responsible for vignetting, difference in exit pupil size depending on film location. With a small aperture on the contrary, the exit pupil size remains equal for all points on the film, increasing the ray passage rate and, hence, slightly increases render time.

Second, the recorded times of the biconvex lens with two refractions are comparatively high, most noticeable in Figure 6.6 in the fluffy cornell box scene. While the low denominator certainly introduces more fluctuation, another factor can be found responsible. Analyzing the amount of dead rays, it was found that this lens, due to its extreme simplicity, has a ray passage rate of virtually 100%. All other, more complex lenses show lower percentages due to only a single exit pupil being calculated to sample. While the percentage difference is small, the additional rays from the biconvex lens require slightly more computations in the scene, hence raising the total time marginally. This effect is, for the same reason as the vignette outliers, most prominent in the complex cornell box scene.

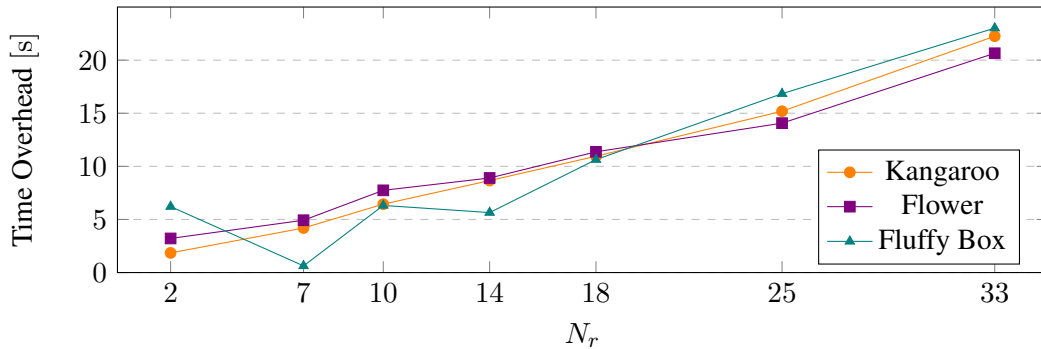


Figure 6.5: Time overhead of different multi-lens camera lenses over the pinhole camera. Lenses are represented by their number of refractions N_r .

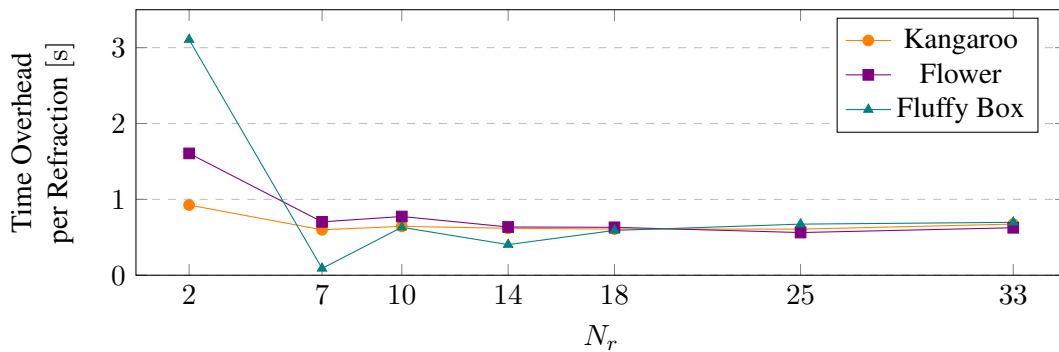


Figure 6.6: Time overhead of different multi-lens camera lenses over the pinhole camera per refraction. Lenses are represented by their number of refractions N_r .

6.3.2 Custom Aperture Shape

Modifying the shape of the aperture not only changes the look of bokeh, but also impacts the total render time. Table 6.3 records render times and the ray passage rate of the three scenes using different aperture shapes on the biconvex lens [Inc21]. The pentagon, star and heart apertures all improve the render time by a large portion, which mainly stems from the low ray passage rate. The heart shape for example drastically limits the aperture region and covers about half of it, consequentially about half of all generated rays hit the aperture and are not processed further. In the case of the 50-edge-polygon aperture, as it closely resembles a circle, a virtually flawless ray passage rate is achieved. Overall though, the ray passage rate is slightly lower than the average relative time, as both the dead rays and the more complex intersection computation occupies some additional time. In the case of the polygon, this computation complexity depends on the polygon edges, as they have to be looped for each ray. This factor, however, has only marginal impacts, a probable reason is that the C++ compiler can optimize such loops in very effective manners. With the low ray passage rate of some aperture shapes comes another consequence, namely less exposure. This is visible in Figure 6.7, where the renders with the different aperture shapes are displayed side by side. As the ray passage rate determines the amount of rays that contribute to illumination, a direct relationship between it and the brightness of the render is observable.

Applying a custom aperture to other lenses shows similar tendencies. The factor by which the performance changes is, however, not constant, but depends on the setup of the lens. A lens with its aperture close to the film plane stops rays early, whereas a lens with many elements between film and aperture spends more time on the calculation of rays that end up dead. In all measurements with restrictive apertures though, visible in Appendix B, the total render time is reduced drastically and the resulting image is darker due to the lower ray passage rate.

Aperture Shape	Render Time [s]			Average Relative Time [%]	Ray Passage Rate [%]
	Kangaroo	Flower	Fluffy Box		
Round	7.75	20.31	116.38	100.00	100.00
5 Edges	6.42	15.45	88.46	78.29	76.62
50 Edges	8.14	21.58	116.68	103.86	99.97
Star Shape	4.26	9.57	50.73	48.56	45.22
Heart Shape	4.46	9.87	53.23	50.63	47.02

Table 6.3: Render times with different aperture shapes using a biconvex lens. The second column from the right stores the average render time of the three scenes relative to their round aperture benchmark time.

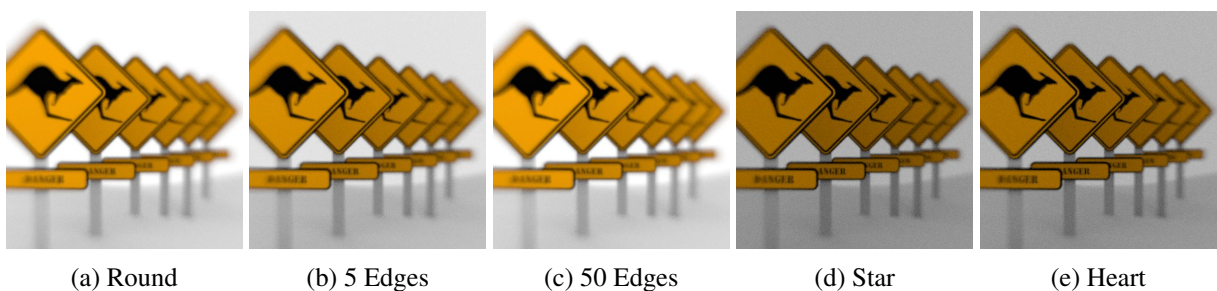


Figure 6.7: Comparison of the rendered images of the kangaroo scene using different aperture shapes on a biconvex lens.

6.3.3 Ray Differentials

Looking at the numbers of Table 6.2, render times of the multi-lens camera are rather high compared to thin lens and pinhole speeds, especially when using complex lenses. Responsible for a large part of this discrepancy is the fact that applesed computes ray differentials for each ray. This is done by supplying partial derivatives with each film point passed to the camera, which are essentially small offsets in x and y direction of the size of half a pixel that are used to perform anti-aliasing. These offsets yield two more starting points for the rays to be shot into the scene. As the three points differ, albeit only slightly, while the sampled exit pupil point remains the same, each ray differs in both origin and direction. All three rays have thus to be traced all the way through the lens individually. This triples the amount of lens tracing computations and consequentially triples the overhead over the pinhole camera.

In the same fashion as in Table 6.2, render times were recorded for the same lenses with ray differentials turned off and can be found in Appendix B. In terms of image quality, no difference is observable with the naked eye, the renders are therefore not additionally included in Appendix C. Displaying the difference of the time overhead over the pinhole camera with and without ray differentials graphically yields Figure 6.8. Once again, a linear increase in time difference with increasing lens complexity can be observed. Additionally, all scenes have roughly the same time difference, independent of the scene's complexity. This meets the expectations, as the number of spawned rays and, hence, the required time for tracing rays through the lens remains the same across all three scenes. Additionally, when computing the relative time overhead for both with and without ray differentials, the overhead with ray differentials is on average three times higher than the overhead without ray differentials. This validates the above claim that the overhead is threefold when rendering with ray differentials.

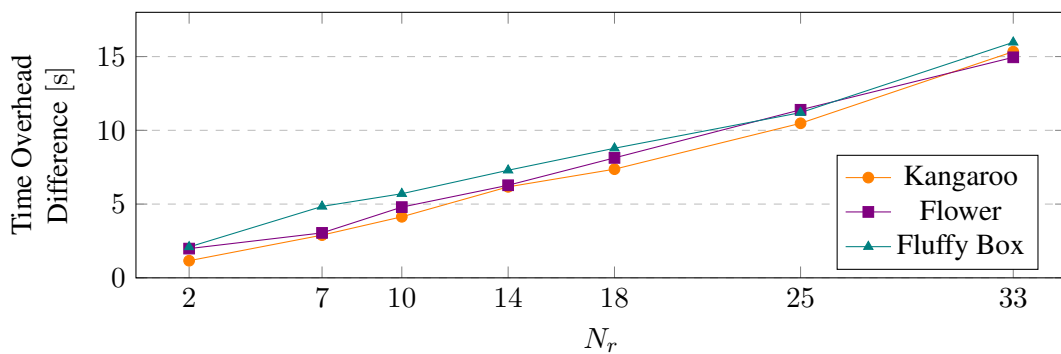


Figure 6.8: Difference of pinhole time overhead with and without ray differentials of different lenses, represented by their number of refractions N_r .

7 Conclusion and Future Work

The open source rendering engine appleseed uses a camera to spawn rays, which are then used in ray-tracing to generate a photorealistic image. The most popular camera model, the pinhole camera, is rather limited in its abilities to achieve realism, its simplicity and the resulting speed make it a viable option for many applications nonetheless. To address the shortcomings of the pinhole camera, the thin and thick lens models feature a single lens, which can offer depth of field and selective focus. These camera models are based on first-order approximations of image formation formulae and are able to achieve perfect imagery, often described as Gaussian optics. To achieve true photorealism, higher-order terms are required, as they introduce aberrations, of which the five Seidel aberrations are most central.

To achieve Seidel aberrations as well as other camera lens effects, a new camera model is needed that simulates the multi-lens structure of modern photographic lenses. Based on the findings of research, a comparison of the two main approaches and of the models within was conducted. It turned out that most papers do not feature overarching comparisons, especially when it comes to performance. While ray simulation requires additional computations in the lens, polynomial optics has to rebuild the polynomial from scratch each time the lens changes. In terms of image quality and achievable effects, different models could achieve different effects and no clear advantage could be perceived for any of the two approaches. In the end, the decision fell on ray simulation, as its physically-based nature fits better into appleseed, both logically and implementation-wise, and tunable lens properties are less costly when virtually no precomputations are required. Admittedly though, both approaches have similar potential and for many, the choice is down to a question of faith.

The new multi-lens camera model based on ray simulation was then contributed to appleseed's open source repository. It features full parameterizability, like a photographer would be able to on his camera, including focal length, f-number and focus distance. Appleseed users specify a lens description file, either self-made or from existing sources, which can range from a single lens to sophisticated modern designs and feature all sorts of characteristics, be it wide angle fisheye lenses or very narrow telephoto lenses. As long as the right format is given, no limits are set. To improve performance and improve the ray passage rate through the lens, an exit pupil is first computed following the approach of Wu et al. [Wu+10], at which rays are then directed at. Unlike the thin lens, custom aperture shapes are considered directly in the tracing loop and do not require the workaround of sampling in the specific shape.

By creating suitable scenes in Blender and exporting them to appleseed, the range of achievable effects of the implementation was shown. Besides variable focal length, f-number and focus, it includes all five Seidel aberrations as well as vignetting. As the aberrations and the behavior of the real aperture have large impacts on the look of bokeh, a bokeh scene was compared between the thin lens and multi-lens camera using different aperture shapes. Looking at the renders, the new model creates far more realistic and nuanced bokeh able to convey depth and periphery, compared to uniform and rather dull blur spots of the thin lens camera. By implementing the thin lens's approach of sampling in a specific shape in the multi-lens camera, it could be shown that this approach is inferior to modifying the real aperture. Rendering scenic images involving lights therefore greatly benefits from the multi-lens camera and offers more variety and realism. Next, natural vignetting of the new model was compared to appleseed's post-processing vignetting, with the result that post-processing offers more customizability, but the multi-lens model creates distinct and varying vignetting depending on the chosen lens.

In a second step, performance was analyzed and compared to other camera models. It was shown that the implemented model requires longer render time, but that the overhead is solely dependent on the lens's complexity and constant per computed refraction, no matter the scene's complexity. Without refractions, equal speeds

7 Conclusion and Future Work

as the pinhole camera were measured, which proves that the new model without lens elements is essentially a pinhole camera. Unfortunately, as *appleseed* calculates ray differentials by default, the overhead is thrice as large, making the camera model non-profitable for simple scenes. When rendering complex scenes or forgoing anti-aliasing, the multi-lens camera becomes worthwhile, especially when the scene allows for affection by optical aberrations.

The multi-lens camera model, although not acting as a thin lens camera replacement, offers a number of realism-improving features that prevail over other cameras, especially in a bokeh-related context. Depending on the scene and the desired lens, the user has to put up with a certain performance deficiency, especially when anti-aliasing is desired. As the derived rays closely resemble the main ray, ways could be found to infer one from the other, without having to retrace the whole lens. Such an optimization would greatly improve the performance and make the model viable also for simpler scenes. While the single exit pupil allows for neat vignetting, it can lead to inefficient sampling, bad passage rates and thus to underexposed images, for example when a small custom aperture shape is chosen or when the lens has greatly differing exit pupils for different pixels. There, intelligent ray generation like the method by Zheng and Zheng [ZZ16] could provide improvement. Additionally, a couple of relevant effects are not included in the scope of this topic and remain open for implementation. First and foremost, this is chromatic aberration, the dispersion of rays depending on wavelengths. Wu et al. [WZHX13] describe its implementation in a follow-up paper, where rays are split into separate wavelengths and traced as bundles through the lens. This could certainly be extended by the open-source community around *appleseed* in the future, but would require a redesign of *appleseed*'s rendering pipeline to access wavelengths inside the camera implementation. Further missing is lens flare, for which efficient computation approaches exist that offer flexibility between accuracy and performance [HESL11; LE13]. Beyond that, aperture diffraction [SDHL11; SHD19] or aspheric lens support [WZHL11; Joo+16] remain potential enhancements.

Bibliography

- [Ang55] Pierre Angenieux. Angenieux. United States Patent 2701982. Feb. 1955.
- [Bar+03] Brian A. Barsky, Daniel R. Horn, Stanley A. Klein, Jeffrey A. Pang and Meng Yu. Camera Models and Optical Systems Used in Computer Graphics: Part I, Object-Based Techniques. In: *Proceedings Computational Science and Its Applications*. Berlin, Heidelberg: Springer, 2003, pp. 246–255. ISBN: 978-3-540-44842-6. DOI: 10.1007/3-540-44842-X_26.
- [Bea+19] François Beaune, Esteban Tovagliari, Luis Barrancos, Stephen Agyemang, Sagnik Basu, Mandeep Bhutani, Lovro Bosnar, Iafael Brune, Matt Chan, João Marcos Mororo Costa, Herbert Crepaz, Junchen Deng, Jonathan Dent, Mayank Dhiman, Dorian Fevrier, Karthik Ramesh Iyer, Dibyadwati Lahiri, Kevin Masson, Gray Olson, Achal Pandey, Jino Park, Sergo Pogosyan, Bassem Samir, Oleg Smolin, Thibault Vergne, Luke Wilimitis and Lars Zawallich. appleseed. Version 2.1.0-beta. Sept. 2019. DOI: 10.5281/zenodo.3456967.
- [BW99] Max Born and Emil Wolf. Principles of Optics: Electromagnetic Theory of Propagation, Interference and Diffraction of Light. 7th ed. Cambridge University Press, 1999. ISBN: 978-1-139-64418-1. DOI: 10.1017/CBO9781139644181.
- [Bre58] Theodor Brendel. Three-Membered Photographic Objective. United States Patent 2854889. Agfa-Gevaert N.V. Oct. 1958.
- [CPC84] Robert L. Cook, Thomas Porter and Loren Carpenter. Distributed Ray Tracing. In: *Proceedings ACM SIGGRAPH Computer Graphics*. Vol. 18. 3. July 1984, pp. 137–145. DOI: 10.1145/964965.808590.
- [Fra03] W. Randolph Franklin. PNPOLY - Point Inclusion in Polygon Test. 2003. URL: https://wrf.ecse.rpi.edu/Research/Short_Notes/pnpoly.html (visited on 26th May 2021).
- [Gau41] Carl Friedrich Gauss. Dioptrische Untersuchungen. Göttingen: Dieterich, 1841.
- [HD14] Johannes Hanika and Carsten Dachsbacher. Efficient Monte Carlo Rendering with Realistic Lenses. In: *Computer Graphics Forum* 33.2 (Apr. 2014), pp. 323–332. DOI: 10.1111/cgf.12301.
- [Hec17] Eugene Hecht. Optics. 5th ed. Pearson Education, 2017. ISBN: 978-0-13-397722-6.
- [HSS97] Wolfgang Heidrich, Philipp Slusallek and Hans-Peter Seidel. An Image-Based Model for Realistic Lens Systems in Interactive Computer Graphics. In: *Proceedings CIPS Graphics Interface*. 1997, pp. 68–75.
- [Hud68] Lena M. Hudson. Modified 4-Member Gaussian Projection Objective. United States Patent 3376090. Bausch and Lomb Inc. Apr. 1968.
- [HESL11] Matthias Hullin, Elmar Eisemann, Hans-Peter Seidel and Sungkil Lee. Physically-Based Real-Time Lens Flare Rendering. In: *ACM Transactions on Graphics* 30.4 (July 2011), pp. 1–10. DOI: 10.1145/2010324.1965003.
- [HHH12] Matthias Hullin, Johannes Hanika and Wolfgang Heidrich. Polynomial Optics: A Construction Kit for Efficient Ray-Tracing of Lens Systems. In: *Computer Graphics Forum* 31.4 (2012), pp. 1375–1383. DOI: 10.1111/j.1467-8659.2012.03132.x.

Bibliography

- [Ike82] Keiji Ikemori. Retro-Focus Type Wide Angle Lens. United States Patent 4310222. Canon Inc. Jan. 1982.
- [Inc21] Thorlabs Inc. N-BK7 Bi-Convex Lenses, Uncoated. 2021. URL: https://www.thorlabs.com/newgrouppage9.cfm?objectgroup_id=4847 (visited on 8th July 2021).
- [Ito80] Takayuki Itoh. Wide Angle Zoom Lens System. United States Patent 4196968. Pentax Corporation. Apr. 1980.
- [Joo+16] Hyuntae Joo, Soonhyeon Kwon, Sangmin Lee, Elmar Eisemann and Sungkil Lee. Efficient Ray Tracing Through Aspheric Lenses and Imperfect Bokeh Synthesis. In: *Computer Graphics Forum* 35.4 (2016), pp. 99–105. DOI: 10.1111/cgf.12953.
- [Kin92] Rudolph Kingslake. Optics in Photography. Bellingham, Washington: SPIE Press, 1992. ISBN: 978-1-62841-379-3.
- [KMH95] Craig Kolb, Don Mitchell and Pat Hanrahan. A Realistic Camera Model for Computer Graphics. In: *Proceedings ACM SIGGRAPH Computer Graphics and Interactive Techniques*. New York, 1995, pp. 317–324. DOI: 10.1145/218380.218463.
- [Lai95] Milton Laikin. Lens Design. 2nd ed. Optical Science and Engineering. New York: Marcel Dekker Inc., 1995. ISBN: 978-0-8247-9602-0.
- [LE13] Sungkil Lee and Elmar Eisemann. Practical Real-Time Lens-Flare Rendering. In: *Computer Graphics Forum* 32.4 (July 2013), pp. 1–6. DOI: 10.1111/cgf.12145.
- [Mah11] Virendra N. Mahajan. Aberration Theory Made Simple. 2nd ed. SPIE Press, 2011. ISBN: 978-0-8194-8825-1. DOI: 10.1117/3.903924.
- [Miz07] Keiko Mizuguchi. Fisheye Lens. United States Patent 7161746. Nikon Corporation. Jan. 2007.
- [Nas10] H.H. Nasse. Depth of Field and Bokeh. In: *Carl Zeiss - Camera Lens Division March* (2010).
- [Oga96] Hideki Ogawa. Zoom Lens. United States Patent 5537259. Canon Inc. July 1996.
- [PP93] Frank L. Pedrotti and Leno S. Pedrotti. Introduction to Optics. 2nd ed. Prentice Hall, 1993. ISBN: 978-0-13-501545-2.
- [PJH16] Matt Pharr, Wenzel Jakob and Greg Humphreys. Physically Based Rendering: From Theory to Implementation. 3rd ed. San Francisco: Morgan Kaufmann Publishers Inc., 2016. ISBN: 978-0-12-800645-0.
- [PC82] Michael Potmesil and Indranil Chakravarty. Synthetic Image Generation with a Lens and Aperture Camera Model. In: *ACM Transactions on Graphics* 1.2 (1982), pp. 85–108. DOI: 10.1145/357299.357300.
- [SHD16] Emanuel Schrade, Johannes Hanika and Carsten Dachsbacher. Sparse High-Degree Polynomials for Wide-Angle Lenses. In: *Computer Graphics Forum* 35.4 (2016), pp. 89–97. DOI: 10.1111/cgf.12952.
- [SHD19] Emanuel Schrade, Johannes Hanika and Carsten Dachsbacher. Brute-Force Calculation of Aperture Diffraction in Camera Lenses. Tech. rep. Karlsruhe Institute of Technology, 2019.
- [Sei57] Philipp Ludwig Seidel. Ueber die Theorie der Fehler, mit welchen die durch optische Instrumente gesehenen Bilder behaftet sind, und über die mathematischen Bedingungen ihrer Aufhebung. In: *Abhandlungen der Naturwissenschaftlich-Technischen Commission bei der Königl. Bayerischen Akademie der Wissenschaften in München* (1857).
- [Smi92] Warren. J. Smith. Modern Lens Design: A Resource Manual. McGraw-Hill, 1992. ISBN: 978-0-07-143830-8.

Bibliography

- [Smi00] Warren. J. Smith. *Modern Optical Engineering*. 3rd ed. McGraw-Hill, 2000. ISBN: 978-0-07-136360-0.
- [SDHL11] Benjamin Steinert, Holger Dammertz, Johannes Hanika and Hendrik P. A. Lensch. General Spectral Camera Lens Simulation. In: *Computer Graphics Forum* 30.6 (Sept. 2011), pp. 1643–1654. DOI: 10.1111/j.1467-8659.2011.01851.x.
- [Thr94] E. Broydrick Thro. Leonardo da Vinci’s Solution to the Problem of the Pinhole Camera. In: *Archive for History of Exact Sciences* 48.3/4 (1994), pp. 343–371. DOI: 10.1007/BF00375086.
- [Tro54] Albrecht Wilhelm Tronnier. Photographic Objective Comprising Four Lens Members Separated by Air Spaces and Enclosing the Diaphragm. United States Patent 2673491. Voigtlander AG. Mar. 1954.
- [Vea97] Eric Veach. *Robust Monte Carlo Methods for Light Transport Simulation*. PhD thesis. Stanford, 1997. ISBN: 978-0-591-90780-3.
- [Vor21] Todd Vorenkamp. Understanding Bokeh. In: *B&H Explora* (July 2021). URL: <https://www.bhphotovideo.com/explora/photography/tips-and-solutions/understanding-bokeh> (visited on 12th Aug. 2021).
- [Wel91] Walter T. Welford. *Useful Optics*. Chicago Lectures in Physics. Chicago: University of Chicago Press, 1991. ISBN: 978-0-226-89305-1.
- [WZHL11] Jiaze Wu, Changwen Zheng, Xiaohui Hu and Chao Li. An Accurate and Practical Camera Lens Model for Rendering Realistic Lens Effects. In: *Proceedings IEEE Computer-Aided Design and Computer Graphics*. Sept. 2011, pp. 63–70. DOI: 10.1109/CAD/Graphics.2011.18.
- [Wu+10] Jiaze Wu, Changwen Zheng, Xiaohui Hu, Yang Wang and Liqiang Zhang. Realistic Rendering of Bokeh Effect Based on Optical Aberrations. In: *The Visual Computer* 26.6-8 (June 2010), pp. 555–563. DOI: 10.1007/s00371-010-0459-5.
- [WZHX13] Jiaze Wu, Changwen Zheng, Xiaohui Hu and Fanjiang Xu. Rendering Realistic Spectral Bokeh Due to Lens Stops and Aberrations. In: *The Visual Computer* 29.1 (Jan. 2013), pp. 41–52. DOI: 10.1007/s00371-012-0673-4.
- [You89] Matt Young. The Pinhole Camera: Imaging Without Lenses or Mirrors. In: *The Physics Teacher* 27.9 (Dec. 1989), pp. 648–655. DOI: 10.1119/1.2342908.
- [ZZ16] Quan Zheng and Changwen Zheng. Adaptive Light Paths Generation Through Full Lens Model. In: *Proceedings IEEE Natural Computation, Fuzzy Systems and Knowledge Discovery*. Aug. 2016, pp. 1846–1851. DOI: 10.1109/FSKD.2016.7603459.
- [ZZ17a] Quan Zheng and Changwen Zheng. Adaptive Sparse Polynomial Regression for Camera Lens Simulation. In: *The Visual Computer* 33.6-8 (2017), pp. 715–724. DOI: 10.1007/s00371-017-1402-9.
- [ZZ17b] Quan Zheng and Changwen Zheng. NeuroLens: Data-Driven Camera Lens Simulation Using Neural Networks. In: *Computer Graphics Forum* 36.8 (2017), pp. 390–401. DOI: 10.1111/cgf.13087.

Appendix A Lens Designs

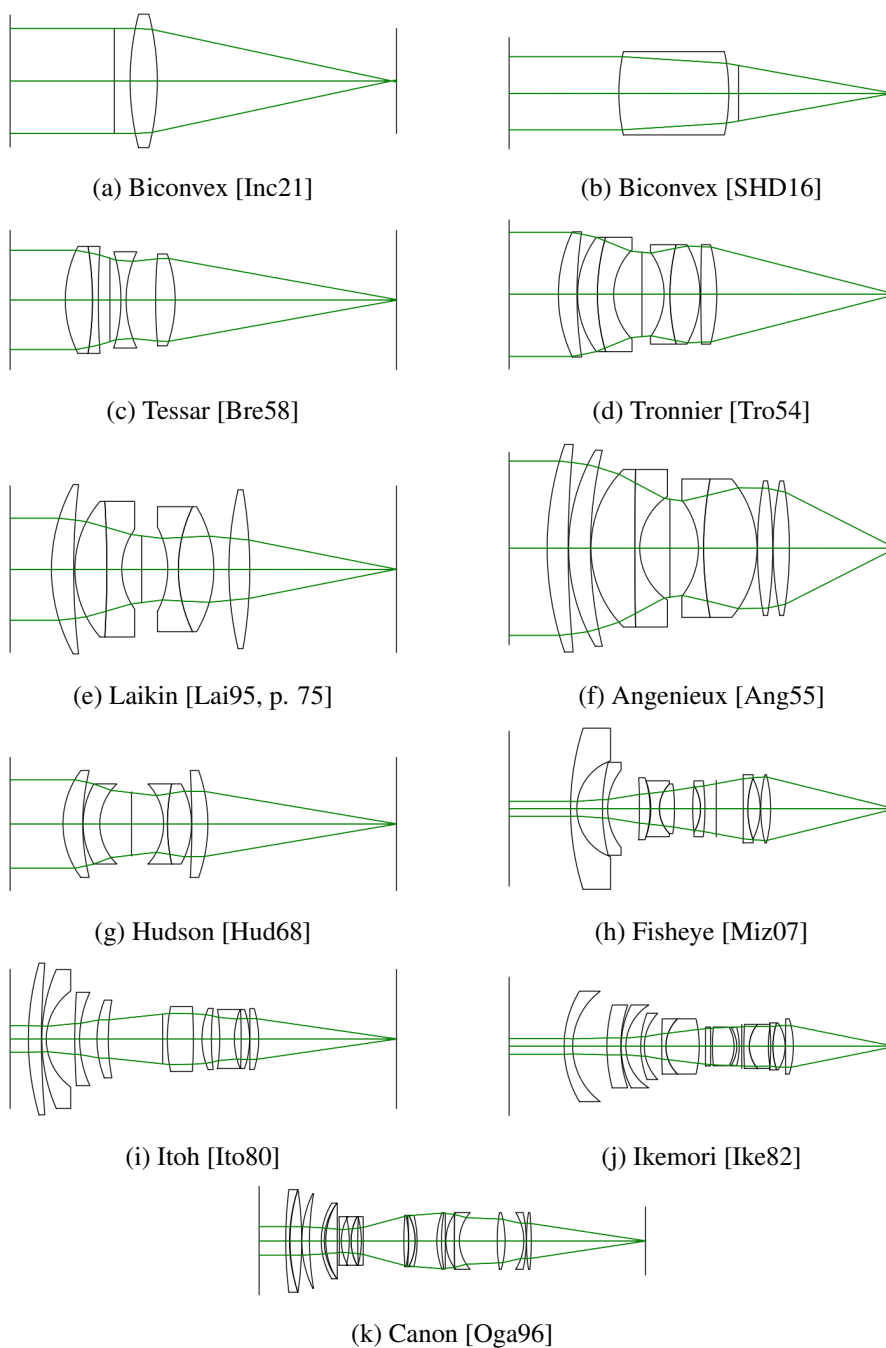


Figure A.1: Designs of all lenses used in Chapters 5 and 6. Visualizations were created using RayOpt⁹.

⁹<https://github.com/quartiq/rayopt> (visited on 22nd August 2021)

Appendix B Performance Measurements

Lens	Run 1	Run 2	Run 3	Average
Pinhole	5.874	5.881	5.936	5.8970
Thin Lens	6.384	6.348	6.368	6.3667
Aperture	6.054	5.987	6.047	6.0293
Biconvex	6.582	6.606	6.571	6.5863
Tessar	7.213	7.155	7.218	7.1953
Tronnier	8.214	8.164	8.218	8.1987
Angenieux	8.409	8.396	8.363	8.3893
Itoh	9.478	9.515	9.452	9.4817
Ikemori	10.627	10.428	10.770	10.6083
Canon	12.652	12.611	13.177	12.8133
Aperture [*]	6.445	6.362	6.378	6.3950
Biconvex [*]	7.721	7.718	7.804	7.7477
Tessar [*]	10.018	10.067	10.187	10.0907
Tronnier [*]	12.392	12.383	12.236	12.3370
Angenieux [*]	14.151	14.179	15.343	14.5577
Itoh [*]	17.060	16.746	16.736	16.8473
Ikemori [*]	20.810	20.775	21.672	21.0857
Canon [*]	28.189	27.997	28.256	28.1473
Aperture, 5 Edges [*]	5.133	5.133	5.151	5.1390
Biconvex, 5 Edges [*]	6.458	6.387	6.402	6.4157
Biconvex, 50 Edges [*]	8.167	8.125	8.136	8.1427
Biconvex, Star [*]	4.264	4.259	4.260	4.2610
Biconvex, Heart [*]	4.471	4.447	4.461	4.4597
Angenieux, 5 Edges [*]	11.520	11.735	11.522	11.5923
Canon, 5 Edges [*]	21.199	21.237	22.542	21.6593

^{*} Includes ray differentials for anti-aliasing.

Table B.1: All measured rendering times of the kangaroo scene at 100 samples per pixel, 50 mm focal length and, where applicable, $f/4$ and focus at 1 m.

Appendix B Performance Measurements

Lens	Run 1	Run 2	Run 3	Average
Pinhole	16.550	17.113	17.644	17.1023
Thin Lens	16.975	17.159	17.754	17.2960
Aperture	16.944	17.277	17.526	17.2490
Biconvex	18.347	18.288	18.360	18.3317
Tessar	19.006	18.760	19.177	18.9810
Tronnier	19.959	19.868	20.319	20.0487
Angenieux	19.900	19.818	19.472	19.7300
Itoh	20.500	20.359	20.158	20.3390
Ikemori	19.470	19.395	20.467	19.7773
Canon	23.371	22.535	22.497	22.8010
Aperture*	17.147	17.725	17.189	17.3537
Biconvex*	20.286	19.810	20.846	20.3140
Tessar*	21.958	21.943	22.183	22.0280
Tronnier*	24.766	24.456	25.310	24.8440
Angenieux*	26.094	26.598	25.327	26.0063
Itoh*	28.484	27.735	29.194	28.4710
Ikemori*	31.049	30.703	31.757	31.1697
Canon*	38.496	37.220	37.537	37.7510
Aperture, 5 Edges*	13.288	13.396	13.278	13.3207
Biconvex, 5 Edges*	15.270	15.250	15.835	15.4517
Biconvex, 50 Edges*	21.048	21.669	22.019	21.5787
Biconvex, Star*	9.526	9.476	9.701	9.5677
Biconvex, Heart*	9.874	9.846	9.886	9.8687
Angenieux, 5 Edges*	19.308	19.630	20.556	19.8313
Canon, 5 Edges*	30.815	29.441	30.515	30.2570

* Includes ray differentials for anti-aliasing.

Table B.2: All measured rendering times of the flower scene at 100 samples per pixel, 50 mm focal length and, where applicable, $f/4$ and focus at 1.2 m.

Appendix B Performance Measurements


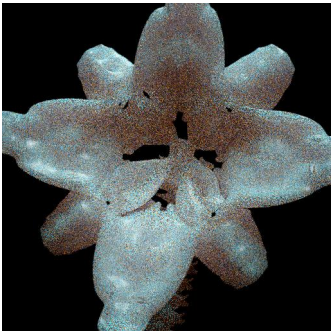
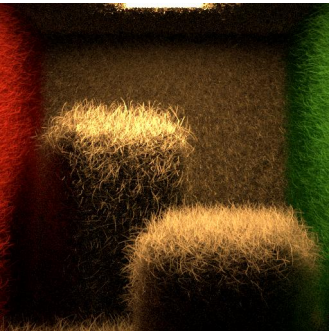

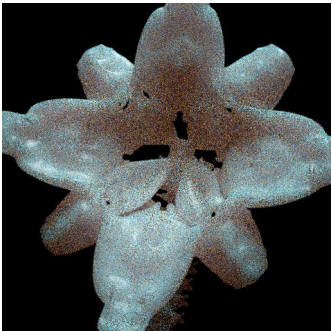
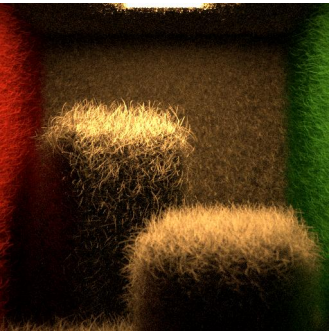

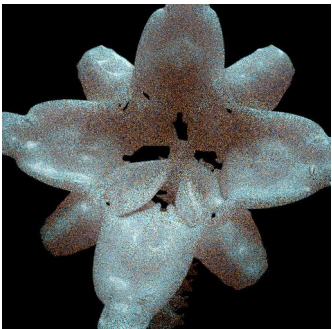
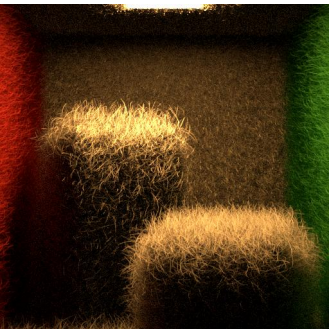
Lens	Run 1	Run 2	Run 3	Average
Pinhole	110.179	110.376	109.961	110.1720
Thin Lens	110.349	111.133	111.038	110.8400
Aperture	109.143	110.028	108.483	109.2180
Biconvex	115.140	113.807	113.925	114.2907
Tessar	106.159	106.084	105.633	105.9587
Tronnier	110.781	111.395	110.189	110.7883
Angenieux	108.223	108.826	108.527	108.5253
Itoh	110.860	113.129	112.073	112.0207
Ikemori	115.608	116.387	115.443	115.8127
Canon	117.994	118.076	115.595	117.2217
Aperture*	109.427	112.032	110.375	110.6113
Biconvex*	116.680	116.775	115.698	116.3843
Tessar*	110.038	111.272	111.092	110.8007
Tronnier*	116.391	116.727	116.356	116.4913
Angenieux*	115.754	116.498	115.206	115.8193
Itoh*	120.604	121.178	120.635	120.8057
Ikemori*	126.571	127.175	127.294	127.0133
Canon*	132.398	132.858	134.307	133.1877
Aperture, 5 Edges*	83.388	84.143	83.641	83.7240
Biconvex, 5 Edges*	88.204	88.576	88.612	88.4640
Biconvex, 50 Edges*	116.520	116.673	116.837	116.6767
Biconvex, Star*	50.625	51.229	50.349	50.7343
Biconvex, Heart*	52.811	53.444	53.447	53.2340
Angenieux, 5 Edges*	89.068	89.501	89.249	89.2727
Canon, 5 Edges*	102.761	103.216	102.980	102.9857

* Includes ray differentials for anti-aliasing.

Table B.3: All measured rendering times of the fluffy cornell box scene at 100 samples per pixel, 50 mm focal length and, where applicable, $f/4$ and focus at 1.1 m. The building of the initial BVH tree is excluded.

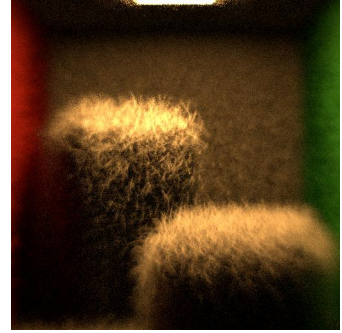
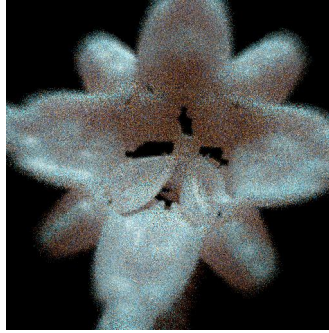
Appendix C Performance Renders

Table C.1: All renders of the performance comparison rendered at 100 samples per pixel including ray differentials, 50 mm focal length and, where applicable, $f/4$.

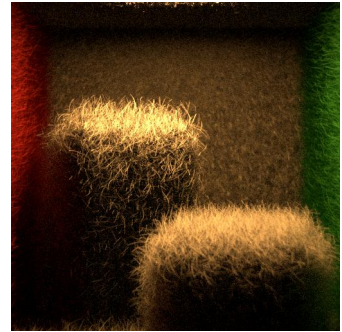
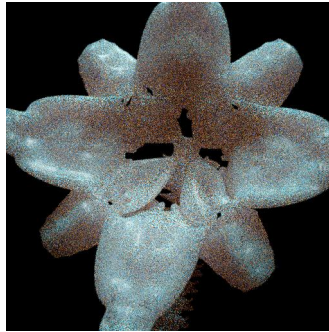
	Kangaroo Scene	Flower Scene	Fluffy Cornell Box Scene
Pinhole			
Thin lens			
Aperture			

Appendix C Performance Renders

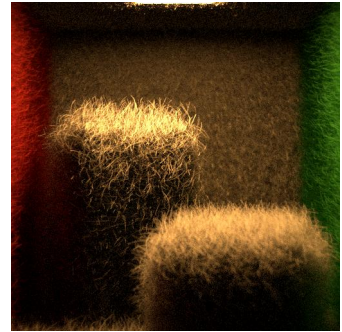
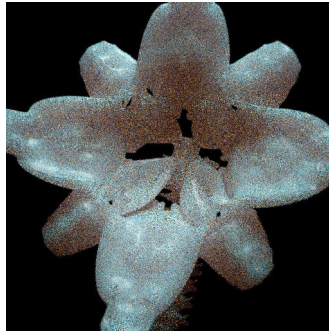
Biconvex



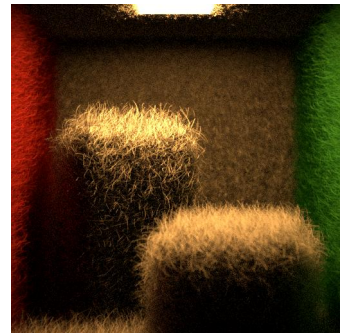
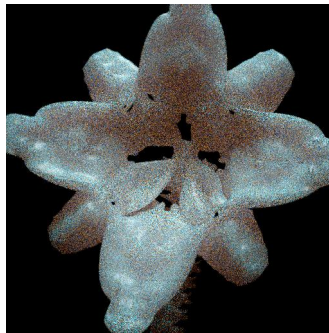
Tessar



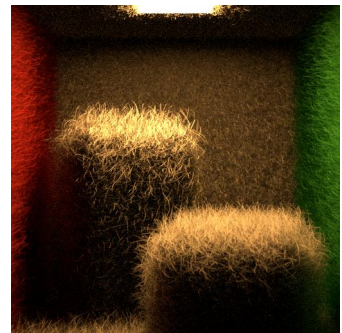
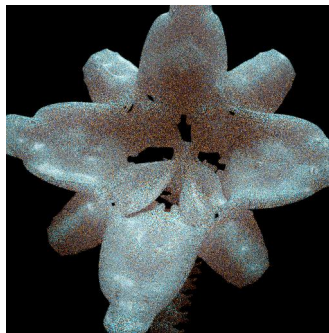
Tronnier



Angenieux

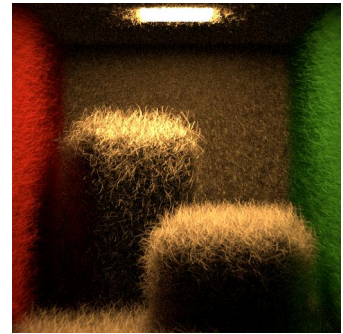
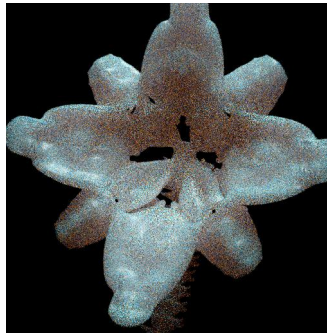


Itoh

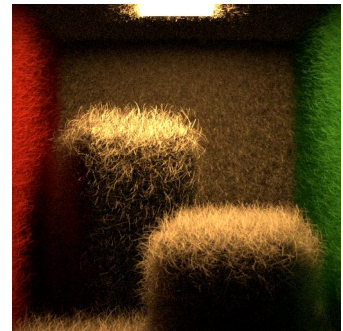
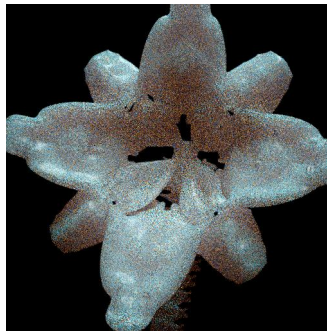


Appendix C Performance Renders

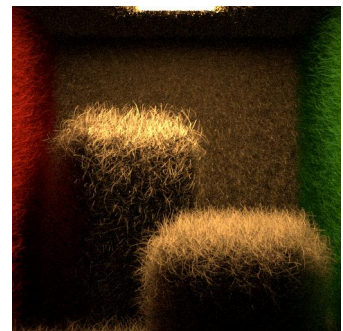
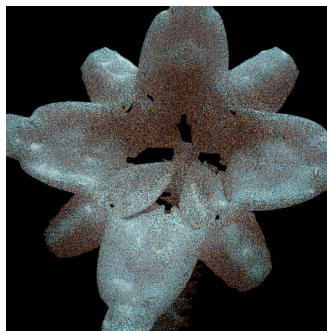
Ikemori



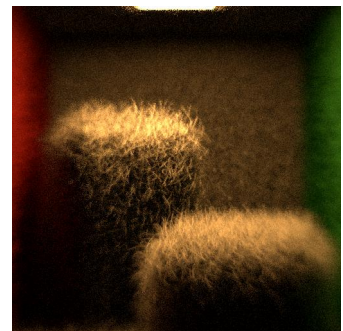
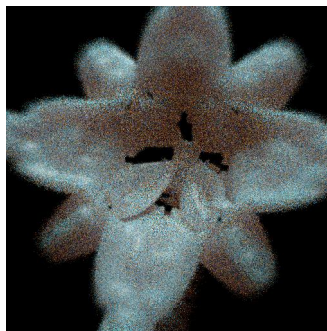
Canon



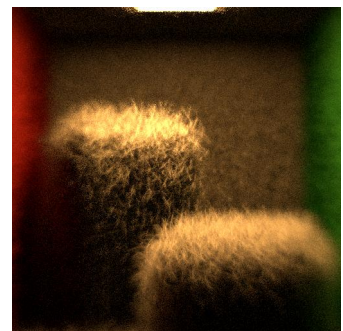
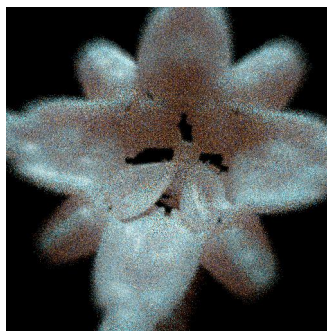
Aperture
5 Edges



Biconvex
5 Edges

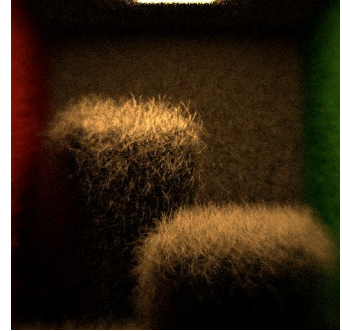
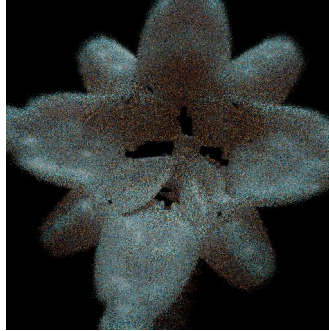


Biconvex
50 Edges

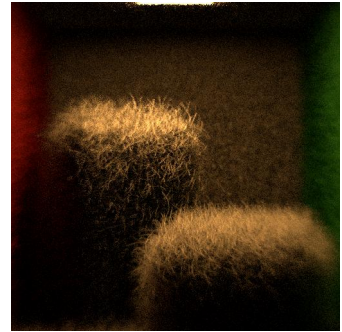
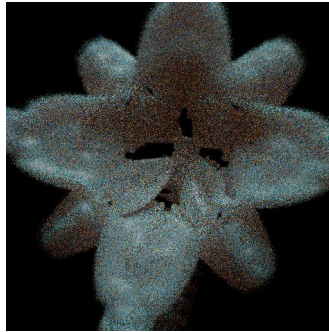


Appendix C Performance Renders

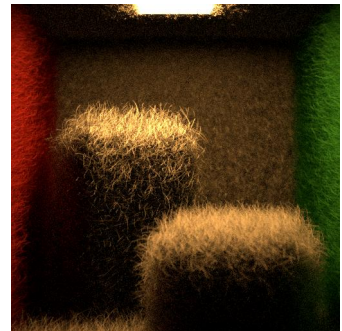
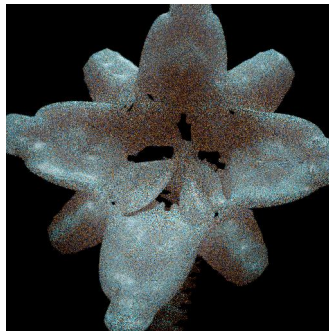
Biconvex
Star



Biconvex
Heart



Angenieux
5 Edges



Canon
5 Edges

