# A Realistic Camera Model for Real-time Rendering

by

**Okka Kyaw**

**PROJECT**

Presented to the Faculty of the
Golisano College of Computer and Information Sciences
Rochester Institute of Technology
in Partial Fulfillment
of the Requirements
for the Degree of

**Master of Science**

**Rochester Institute of Technology**
May 2013

# A Realistic Camera Model for Real-time Rendering

APPROVED BY

SUPERVISING COMMITTEE:

Dr. Joe Geigel, Chair

Dr. Reynold Bailey, Reader

Professor Sean Strout, Observer

## Abstract

# A Realistic Camera Model for Real-time Rendering

Okka Kyaw
Rochester Institute of Technology, 2013

Chair: Dr. Joe Geigel

Traditionally, computer graphics have used a simplified camera model that produces a “perfect” image –a first order approximation of the output from a real camera. Raytracers that accurately simulate physical lens systems have been around for a few decades now, but although modern GPUs are nearing the point where GPU-assisted raytracing can be performed in real-time, they are not fast enough to run on average consumer hardware. On the other hand, many real-time applications have been able to include more sophisticated camera models, simulating specific effects such as depth of field with bokeh and motion blur. However, many of them forego accuracy, simply basing their calculations on what looks plausible.

This project describes a camera model for real-time computer graphics that balances performance with accuracy. Given a set of parameters for the configuration of a real camera, the model simulates the captured image using an analytical approach. Lens and exposure equations from photographic optics are applied in a post-processing step to the output from a rasterizer to create a more physically accurate result. Methods for simulating third order optical aberrations through post-processing are also discussed. As a standard graphics API was used for the renderer, the techniques described here should be easy to integrate into existing real-time graphics engines.

# Table of Contents

# List of Figures

# Chapter 1. Introduction

There are three key components in a renderer: reflection models that simulate how different materials interact with light, illumination algorithms that simulate light transport throughout the 3D scene, and a camera model that simulates image formation and capture. After Cook *et al.* pioneered the simulation of depth of field and motion blur with their distributed raytracer, very little progress was made in the field of camera simulation until the work by Kolb *et al.* on a physically-based camera model for computer graphics[KMH95].

There are many reasons why a realistic camera model is desirable. Kolb *et al.*[KMH95] mention a few that are particularly relevant for real-time computer graphics:

- When merging computer-generated imagery with recorded imagery (e.g. for augmented reality or special effects), it is important that the computer-generated imagery uses a camera model similar to that of the real camera.
- A camera metaphor makes using 3D graphics systems easier for users who are already familiar with cameras, and the principles behind 3D graphics are easier to explain when related to real cameras.

## 1.1 Agenda

When modeling a real camera, it is necessary to understand the optical principles related to their design and operation. First, differences in camera simulation in the two main rendering paradigms (rasterization vs. raytracing) are discussed, along with previous work in each field. Next is an overview of the components in a camera and how they can be analyzed using optics theory. As the field of optics is extremely vast, the focus is on topics that are relevant when working with photographic media. Then we describe our camera model and the visual result of different parameters for each camera component. Finally, the output produced by an implementation of our model is shown and discussed.

# Chapter 2. Previous Work

## 2.1 Raytracing

Raytracing provides a numerical solution to the problem of simulating a lens system and the resulting exposure. The correct physical output is integrated by tracing the path of rays traveling through a camera lens system and aggregating the results.

This technique is relatively simple to implement (especially if it is being added on to an existing raytracer) as all it requires is implementing refraction through the elements in the lens, and adds a small, constant amount to the rendering time. The solution also approaches physical accuracy as the number of samples increases. Kolb *et al.* describe such a physically based camera model that served as the inspiration for this project[KMH95].

However, performing the raytracing itself is slow, as the source image has to be obtained by checking ray intersections with the scene. Even the fastest implementations from within the past year run at around 30 frames per second in the best case. This making them unsuitable for use in interactive applications with strict performance requirements, such as game engines, for the time being[LES10].

## 2.2 Post-processing

On the other hand, post-processed approaches to camera simulation try to find an analytical solution by applying lens and optics equations. Given an input image, they calculate the modifications that need to be applied to an image to make it match the expected output.

As an alternative to raytracing, they take the “perfect” output from a rasterizer and add the imperfections to it. The post-processing step can be applied to any input image though, so it is more flexible in that regard. It is also possible to increase or decrease the number and fidelity of the effects being simulated, in order to meet a desired accuracy/performance balance.

However, there is a lot more complexity involved in a post-processed camera simulator, as each individual effect needs to be analyzed and simulated. In addition, a lot of the existing research in this area usually focus on simulating one specific effect, and tend to neglect the issue of making sure effects work well in combination. For example, it can be difficult to simulate both defocus blur and motion blur simultaneously using postprocessing, as traditional implementations of these techniques use the screen-space depth buffer, and the blurring introduced by each effect makes it difficult for them to interact with each other.

A lot of promising research is being done in this area in recent years though. For example, a new technique called stochastic rasterization is attempting to solve the problem of rendering defocus blur and motion blur simultaneously[McG+10][Mun+11][AHA11].

# Chapter 3. Camera Optics

In photography, the main components of a camera are the lens, which gathers and focuses incoming light, the aperture, which controls the angle and amount of light rays passing through, and the film or digital sensor, which records the image chemically or electronically.

## 3.1 Lenses

In optics, a *simple lens* is a lens consisting of only one element (i.e. it has two refracting surfaces). A *compound lens* is made up of two or more lens elements, which are generally aligned along a common axis, the *optical axis*. A lens with a thickness that is negligible compared its focal length is a *thin lens*; otherwise it is a *thick lens*. Lens diagrams traditionally show rays of light traveling from left to right, and distances and parameters are measured from the point where incident rays cross the optical axis, the *optical center*. A *principal focus* of a lens is a point on the optical axis onto which *collimated light* (light whose rays are parallel) is focused[Ray02].

### 3.1.1 Focal Length

The main property of a thin lens is its *focal length* ($f$), the distance from the lens to the on-axis principal focus of incident collimated light. Light can travel through a lens in either direction (i.e. optical systems are reversible), so both front and rear focal lengths can be defined. The lens conjugate equation defines the relationship between the focal length, the lens to object distance ($u$), and the lens to image distance ($v$)

$$\frac{1}{u} + \frac{1}{v} = \frac{1}{f} \tag{3.1}$$

*Conjugate planes* orthogonal to the optical axis are defined by $u$ and $v$, which are *conjugate points*, and the principal focal plane is defined by $f$. When focused at infinity, $f = v$, and to reduce the focal distance (i.e. bring the focus closer), $v$ has to increase. Image magnification ($m$), the magnifying power of the lens, is then defined as

$$m = \frac{v}{u} \tag{3.2}$$

Useful equations for calculating $u$ and $v$ can be derived, and these can be used for making the lens focus at a desired distance[Ray02].

$$u = f\left(1 + \frac{1}{m}\right) \tag{3.3}$$

$$v = f\,(1 + m) \tag{3.4}$$

$$u = \frac{vf}{v - f} \tag{3.5}$$

$$v = \frac{uf}{u - f} \tag{3.6}$$

### 3.1.2 Gaussian Optics

The Taylor series for the sine and cosine functions describe each function as an infinite sum

$$\sin(\phi) = \phi - \frac{\phi^3}{3!} + \frac{\phi^5}{5!} - \cdots$$

$$\cos(\phi) = 1 - \frac{\phi^2}{2!} + \frac{\phi^4}{4!} - \cdots$$

A *paraxial ray* is a ray that forms a small angle with the optical axis and remains close to the axis throughout the system. Image formation that only considers paraxial rays is termed *Gaussian optics*. This technique makes use of the *paraxial approximation*, which is a truncation of the Taylor series to a first-order approximation

$$\sin(\phi) \approx \phi \tag{3.7a}$$

$$\cos(\phi) \approx 1 \tag{3.7b}$$

$$\tan(\phi) = \frac{\sin(\phi)}{\cos(\phi)} \approx \phi \tag{3.7c}$$

The approximation simplifies the analysis of optical systems, which would otherwise have to be performed in three dimensions. Rays can be analyzed by their interactions along the *meridional plane*, the plane that contains the optical axis. In particular, compound lenses can be regarded as a single entity, and a single *effective focal length* $f$ can be defined for the entire system[Ray02].

### 3.1.3 Field of View

The field of view ($W$) for a lens focused at infinity can be calculated as

$$W = 2 \arctan \left( \frac{K}{2f} \right) \tag{3.8}$$

where $K$ is the diagonal length of the film format. This decreases as the lens is focused at closer distances, and $f$ is replaced by the image conjugate distance $v$[Ray02].

# Chapter 4. Exposure

While a camera lens shapes the path taken by light entering the camera, the aperture and shutter control how the light is exposed and captured by a film or sensor. The exposure $H$ can be defined relative to image illumination $E$ and exposure time $t$

$$H = Et \tag{4.1}$$

## 4.1 Aperture

The *aperture stop* in a lens system limits the amount of light it transmits. Modern cameras do this using a device called an *iris diaphragm*, which uses a series of blades to control the aperture size mechanically[LFS10]. The curvature and number of blades defines the iris shape, which influences the shape taken by out-of-focus highlights in the image and affects its aesthetic quality[Ray02].

### 4.1.1 Entrance and Exit Pupils

The aperture stop, when combined with the other elements in the lens, forms two pupils through which light rays appear to enter and leave the lens system. The *entrance pupil* (En) is the image of the aperture stop as seen through the front of the lens, formed by the lens elements in front of the physical aperture. Its center is the *center of perspective* for the lens, and is the location of the vertex of its field of view. Likewise, the *exit pupil* (Ex) is the image of the aperture stop as seen through the back of the lens, formed by the lens elements behind the physical aperture. Its center is the *center of projection* for the lens.

With asymmetrical lens designs, the entrance and exit pupils may differ in size, and the pupil magnification ($P$) can be defined as

$$P = \frac{\text{diameter of Ex}}{\text{diameter of En}} \tag{4.2}$$

When this value is not unity, any exposure calculations using the lens magnification factor $m$ needs to be replaced by $m/P$. Distances for image illumination and exposure calculations should also be measured from the corresponding pupil for accuracy[Ray02].

### 4.1.2 Relative Aperture

The ratio of a lens' aperture diameter to its focal length determines image illumination, and this ratio is called the *f-number* ($N$)[LFS10]. As the aperture diameter grows smaller, the ratio grows larger, so large $f$-numbers correspond with lower image illumination and vice versa.

For compound lenses, the equivalent focal length ($f$) and entrance pupil diameter ($e$) and are used instead, so at infinity focus

$$N = \frac{f}{e} \tag{4.3}$$

If the entrance pupil is not circular, N can be defined in terms of its area $A$

$$N = \frac{f}{2(A/\pi)^{1/2}} \tag{4.4}$$

Note that the amount of light transmitted by the aperture is proportional to $A$, so it will increase or decrease as a factor of $e^2$. Therefore, a decrease in $N$ by a factor of $\sqrt{2}$ will result in double the amount of light being transmitted. $F$-numbers are often given as *relative aperture* values, written as a fraction of $f$. The series $f/0.5$, $f/1$, $f/1.4$, $f/2$, $f/2.8$, $f/4$, $f/5.6$, etc. gives relative aperture values that progressively halve the light transmission[Ray02].

### 4.1.3 Effective Aperture

$N$ is defined for infinity focus, so as the lens' focus is brought closer, the *effective aperture* $N'$ is defined instead, with $f$ replaced by $v$, the image conjugate distance

$$N' = \frac{v}{e} \tag{4.5}$$

This means that exposure time and $f$-number setting will have to be adjusted accordingly to maintain the same exposure[Ray02].

## 4.2 Shutter

The *shutter* is a device that controls how long the film or sensor in the camera is exposed to incoming light. A between-lens shutter is a shutter that is located in the lens, close to the aperture stop. It is usually a diaphragm shutter, using a mechanism with blades similar to an iris diaphragm. During an exposure, a diaphragm shutter opens and stays open for the exposure period, then closes. A focal plane shutter is a shutter that is positioned in front of the film or sensor. Initially, the film/sensor is covered by a metal blade. During an exposure, this blade opens to let light in, then a second blade closes over the film/sensor. The duration of these exposures is called the *shutter speed*.

An ideal shutter would open and close instantaneously, but as mechanical shutters are used by traditional cameras, their motion can affect the resulting image at extremely fast shutter speeds. In particular, slow focal plane shutters expose different sections of the film/sensor at different times, so object or camera motion can cause the subject to appear compressed or stretched. On the other hand, they do not have to be built into every lens unlike diaphragm shutters, and they can achieve faster shutter speeds.

Some digital cameras do not have this problem as the sensor itself is controlled electronically to control exposure time, but image quality can suffer as the additional circuitry takes up space on the sensor[LFS10].

## 4.3 Exposure Photometry

As mentioned earlier, the exposure of a photo is relative to image illumination and exposure time. If one of these is increased, the other needs to be reciprocally decreased to maintain the same exposure level; this is known as *reciprocity* [LFS10].

Geometric analysis of the rays leaving an object point and arriving at an image point allows us to define an equation for on-axis illuminance ($E$) in lux as follows[Ray02]

$$E = \frac{TL}{4N^2} \tag{4.6}$$

where $L$ is the incoming luminance in nits, $T$ is the transmittance of the lens, and $N$ is the relative aperture. From this, the maximum possible value for the aperture can be calculated as an $f$-number of 0.5, when $E = L$ with unit transmittance.

The equation also reveals the relationship between $E$ and $N$ that was described earlier

$$E \propto \frac{1}{N^2} \tag{4.7}$$

and the illuminance ratio for two different $f$-numbers can be given as

$$\frac{E_1}{E_2} = \frac{{N_2}^2}{{N_1}^2} \tag{4.8}$$

A more general form of the equation can be calculated that accounts for the angle $\theta$ at which the principal ray is entering the lens, substitutes $N(1+m)$ for the effective aperture $N'$, incorporates the the pupil magnification $P$ for non-symmetrical lenses, and includes a vignetting factor $V$ related to $\theta$[Ray02]

$$E = \frac{V\pi TLcos^4\theta}{4N^2[1+(m/P)]^2} \tag{4.9}$$

From equation 4.1

$$\frac{H_1}{H_2} = \frac{{N_2}^2}{{N_1}^2} \tag{4.10}$$

and

$$E \propto \frac{1}{t} \tag{4.11}$$

so

$$\frac{E_1}{E_2} = \frac{t_2}{t_1} \tag{4.12}$$

The aperture size and exposure time needed to produce equivalent exposures is then given as

$$\frac{t_1}{t_2} = \frac{{N_1}^2}{{N_2}^2} \tag{4.13}$$

## 4.4 Vignetting

Vignetting refers to the reduction in image brightness or saturation towards the image periphery[LFS10].

The first type, *natural vignetting*, can be seen from equation 4.9 as the *$cos^4\theta$ law of illumination*

$$E \propto cos^4\theta \tag{4.14}$$

The next type, *optical vignetting*, occurs when off-axis light is occluded due to lens elements shading each other. This can be estimated by projecting images of the entrance pupil and rear lens element onto the front element, and taking the area of the pupil that isn't covered by either element. The result is also known as the *cat's-eye effect* due to its shape, and can be mitigated by decreasing the aperture size so that more of the entrance pupil remains within the common area.

The last type, *mechanical vignetting*, can occur when the corners of the image are blocked off by parts of the lens system or lens hood. These parts are rendered out of focus by the lens, but they will still cause shading or obscuring of the affected image regions[Ray02].

## 4.5 Film/Sensor

Photographic film is coated with silver halide salts that react to light and form a visible image when the film is developed. Depending on the specific dyes used along with the silver salts, the film will have a specific response to different wavelengths and intensities of light. This defines the *characteristic curve* for that type of film, a plot of film opacity versus the log of the exposure. Film sensitivity (also known as film speed) is measured using the ISO standard, which grades film based on their overall sensitivity to light. A doubling of the arithmetic ISO value corresponds with a doubling of film sensitivity.

The characteristic curve for the film is mainly composed of a linear portion where the log exposure scales linearly with output brightness. At the upper end (known as the shoulder), it flattens out gradually to compress high luminance values. Likewise, at the lower end (known as the toe), it flattens out gradually to prevent desaturating the blacks. When modeling a characteristic curve to perform tonemapping, the results will be more saturated and may look more visually pleasing compared to standard tone operators like Reinhard[Hab10].

In digital cameras, setting the signal gain of the sensor allows the relationship between exposure and the sensor data values to be arbitrarily defined, and the sensor data values can also be arbitrarily mapped to the output image. Thus there are several standard techniques that camera manufacturers can choose from when measuring the exposure of the different sensitivity settings on a digital camera sensor.

In general, the more sensitive the film or sensor, the more noise will be present in the final image. To increase photographic film sensitivity, larger grains of silver are used so that a larger ratio of particles will be activated given the same number of photons. As this leads to a lower number of particles per unit area, the output will have more fluctuations in optical density and appear graininer. Digital photos do not have grain, but they can still have image noise due to electronic noise in the sensor and circuitry. In addition, they are limited in resolution by the number of cells in the sensor[LFS10].

# Chapter 5. Defocus Blur and Motion Blur

## 5.1 Defocus Blur/Depth of Field

A camera lens can only focus exactly at one distance, but the amount of defocus grows slowly on each side of the focal plane. The blur circle due to defocus is also known as the *circle of confusion*. As the resulting blurriness only becomes noticeable once the circle of confusion is big enough, the distance range within which the image is acceptably sharp is called the *depth of field*[LFS10].

The diameter of the circle of confusion in the image plane ($C$) is related to the point spread function of the lens, which describes how a point source will be imaged by the lens. The geometric size of $C$ (ignoring changes in $C$ due to lens aberrations) can be calculated by starting with the diameter of the circle of confusion in the object space focal plane ($K$). If the focal plane is at distance $u$ from the entrance pupil (En), the defocused object point is at distance $S$ from En, and the entrance pupil diameter is $D$, from similar triangles

$$\frac{K}{|S-u|} = \frac{D}{S}$$

and rearranging gives

$$K = \frac{D|S-u|}{S} \tag{5.1}$$

$D$ can be substituted with effective aperture $N'$ using the relation

$$N' = \frac{v}{D}$$

$$D = \frac{v}{N'}$$

$$K = \frac{v|S-u|}{SN'} \tag{5.2}$$

Magnification $m$ and pupil magnification $P$ can then be applied to $K$ to get $C$

$$C = \frac{mK}{P}$$

$$m = \frac{v}{u}$$

$$C = \frac{v^2|S-u|}{uSN'P} \tag{5.3}$$

For distant scenes, $N'$ can be replaced with just the relative aperture $N$, and for symmetrical lenses P will be unity, so the equation simplifies to

$$C = \frac{v^2|S-u|}{uSN} \tag{5.4}$$

When the aperture is noncircular, the circle of confusion will take on the shape of the aperture stop's image, as mentioned in section 4.1. The aesthetic quality of this blur, which is further influenced by the aberrations in chapter 6, is called *bokeh*[Nas10].

A concept similar to depth of field, the *depth of focus*, affects the image space and determines the tolerance in positioning the image plane[Ray02].

## 5.2 Motion Blur

An ideal camera would be capable of capturing the light in a scene at an instant in time, freezing all motion. However, real cameras have non-zero exposure times, so if an object is moving sufficiently fast enough or the shutter remains open for long enough, areas of the scene that are in motion will appear blurred in the output. Therefore, the sharpness of moving objects in the scene will be affected by shutter speed.

Another possible source of temporal image distortion occurs when a camera has slow focal-plane shutters. As the “curtain” on the shutter moves across the film for the duration of the exposure, a moving object could appear distorted as its movement means it is not exposed evenly across the film[LFS10].

# Chapter 6. Optical Aberrations

There are three conditions that have to be met for a lens to be “perfect,” as defined by the theoretical physicist Maxwell in 1858. Rays from an object point have to meet again at a single image point, an orthogonal object plane has to be imaged as an orthogonal image plane, and object and image shapes have to be identical, only varying in size due to magnification. However, due to the way light interacts with optical systems, the image produced by a lens may deviate from what is predicted by the first-order paraxial approximation. These *aberrations* may be visible in the output image, and their magnitude varies depending on the lens’ focal length, the relative aperture, and the distance from the image point to the optical axis.

There are seven types of aberrations that occur when the lens simulation takes third order factors into account. Five of these are unaffected by the wavelength of light, and are known as *monochromatic aberrations* or (*von*) *Seidel aberrations*. The remaining two are *chromatic aberrations*, varying depending on the wavelength of light passing through the lens. Most photographic lenses are *aberration-limited* in that their performance is limited by the presence of aberrations. However, it is worth noting that even a perfect lens will have to account for diffraction effects due to the wave nature of light, making it a *diffraction-limited* lens[Ray02].

## 6.1 Transverse Ray Aberrations

To calculate parameters describing these third-order aberrations, only three rays need to be traced:

1. An *axial* or *marginal* ray that starts at the axial object point and touches the edge of the pupils and aperture stop.
2. A *principal*, *chief*, or *field* ray that starts from an off-axis point and passes through the center of the pupils and aperture stop.
3. A *skew* ray that starts off-axis in a plane that is not *meridional* (i.e. the plane does not contain the optical axis).

In general, the principal and skew rays will intersect the image plane at different points, and the displacement between these points are the transverse ray aberrations. The skew ray passes through the exit pupil at the polar coordinates ($r$, $\theta$), and the field angle $\beta$ is the angle between the optical axis and the principal ray after it leaves the exit pupil[Ray02].

## 6.2 Wavefront Aberrations

These ray aberrations can be quantified as *wavefront aberrations* ($W$), which describes the amount of deviation in a wavefront after passing through a lens.

$$W = W(\beta, r, \theta) = \sum c_{ABC} \beta^A r^B cos^C \theta \tag{6.1}$$

The order of each term is given as $A + B - 1$, and the equation expands into the following series

$$\begin{aligned} W = {} & W_{020} r^2 + W_{111} \beta r \cos\theta \\ & + W_{040} r^4 + W_{131} \beta r^3 \cos\theta + W_{222} \beta^2 r^2 \cos^2\theta \\ & + W_{220} \beta^2 r^2 + W_{311} \beta^3 r \cos\theta \\ & + \text{higher orders} \end{aligned} \tag{6.2}$$

The first order terms with coefficients $W_{020}$ and $W_{111}$ disappear at the paraxial focus, leaving the next five terms as the third order or *Seidel coefficients*.

## 6.3 Monochromatic Aberrations

The five monochromatic aberrations are *spherical aberration* (SA), *coma*, *astigmatism*, *field curvature*, and *distortion*. SA, coma, and astigmatism cause object points to be imaged as blur patches, failing the first condition for a perfect lens and resulting in an image that is not *stigmatic*. An image that is stigmatic can still fail the other two conditions due to field curvature, which gives a curved image, and distortion, which gives a non-identical image. When corrected for SA and coma, the lens is *aplanatic*, and applying additional correction for astigmatism makes the lens *anastigmatic*.

### 6.3.1 Spherical Aberration

Spherical aberration ($W_{040}r^4$) causes marginal rays to focus closer to the lens than paraxial rays. The diameter of the resulting image blur circle ($D$), with a focal length $f$, relative aperture $N$, and constant $c$ is approximately

$$D \simeq \frac{cf}{N^3} \tag{6.3}$$

As there is no distance of perfect focus, the image plane has to be positioned closer to the lens to attain the best possible focus, where D is minimized [Ray02].

SA causes the circle of confusion to be of nonuniform brightness due to the arrangement of the rays of light. When SA is uncorrected or under-corrected, the circle of confusion will have a dark core surrounded by a bright ring for foreground blur, and a bright core fading out towards the edges for background blur.

The result is that the background will have a softer, more pleasing blur while the foreground will appear harsher and transform lines into double lines.

Over-correction will result in the opposite effect — foreground blur will be softer while background blur will appear harsher[Nas10]. As photographers almost always prefer more pleasing background blur, some lenses have deliberately under-corrected SA (which can often be controlled through the use of floating lens elements), and are sold as soft focus lenses[LFS10].

### 6.3.2 Coma

Coma ($W_{131}\beta r^3 \cos\theta$) occurs when an off-axis point creates a comet-shaped image patch instead of focusing into a point. This occurs due to refraction being different at different *lens zones* (concentric regions of the lens defined based on their distance from the optical axis)[Ray02].

The diameter of the resulting image blur patch ($D$), with a relative aperture $N$, image-plane height $h$, and constant $c$ is

$$D = \frac{ch}{N^2} \tag{6.4}$$

### 6.3.3 Astigmatism

A *meridional plane* (M) is the plane containing both the principal ray and the optical axis, while the *sagittal plane* (S) is orthogonal to M and contains the principal ray but not the optical axis. Rays traveling along M are *tangential rays*, and rays traveling along S are *sagittal rays*. Astigmatism ($W_{222}\beta^2 r^2 \cos^2\theta$) occurs when an off-axis point is imaged as an oval, circle, or line, depending on the location of the film plane. This occurs when tangential rays come to a focus at a different point than the sagittal rays. This can be observed as either the lines oriented in the sagittal direction or the lines oriented in the tangential direction being in focus, but not both at the same time. When uncorrected, the best possible focus would be at the point in between the two foci where the blurring forms the circle of least confusion[Ray02].

The diameter of the resulting image blur patch ($D$), with a relative aperture $N$, focal length $f$, image-plane height $h$, and constant $c$ is

$$D = \frac{ch^2 f}{N} \tag{6.5}$$

### 6.3.4 Field Curvature

Field curvature ($W_{220}\beta^2 r^2$) occurs because a perfect lens that gives perfect, stigmatic imagery forms its image as a paraboloidal *Petzval surface* instead of a flat plane. This causes the Petzval surface to be offset from the image plane with a displacement ($\Delta x$) at image height $h$ as[Ray02]

$$\Delta x = ch^2 \tag{6.6}$$

The diameter of the image blur patch ($D$) due to field curvature, with a relative aperture $N$, focal length $f$, image-plane height $h$, and constant $c$ is

$$D = \frac{ch^2}{fN} \tag{6.7}$$

### 6.3.5 Distortion

Distortion ($W_{311}\beta^3 r \cos\theta$) occurs when the image magnification changes as a cubic function of the distance from the optical axis. Two distinct types of distortion are visible in lenses, *barrel* and *pincushion*. When image magnification decreases with the off-axis distance, it gives the appearance of an image wrapped around a barrel, hence the name. When it increases with the off-axis distance, it is called pincushion distortion.

The percentage ($D$) by which the image point's distance from the axis ($R$) deviates from the expected distance ($r$) is given as

$$D = 100\left(\frac{R-r}{r}\right) \text{ percent} \tag{6.8}$$

This type of distortion, *optical curvilinear distortion*, is distinct from other types of image distortion. *Geometric distortion* occurs due to the lens' angle of view, *perspective distortion* occurs due to the image viewing distance, and *subject distortion* occurs due to relative motion between the camera and subject during exposure[Ray02].

## 6.4 Chromatic Aberrations

The two types of chromatic aberration are *longitudinal chromatic aberration* and *transverse chromatic aberration*, which occur because refractive index varies with wavelength.

Longitudinal chromatic aberration occurs when different wavelengths of light are focused at different distances from the lens. This is visible throughout the entire image but its effect can be reduced by reducing the aperture size. Transverse chromatic aberration occurs when different wavelengths of light are focused at different positions on the focal plane, and become more pronounced with distance from the optical axis[LB11].

Spherical aberration combines with chromatic aberration and results in *spherochromatic aberration*, which is visible as a colored fringe around spherical aberration[Ray02].

# Chapter 7. Implementation Details

The program is written as a standard Windows desktop application using C++ and DirectX with HLSL shaders.

## 7.1 Choice of Units

Note that lighting calculations can be performed using either photometric or radiometric units, as long as proper conversion is done between the two using the luminosity function. In this implementation, photometric units are used as that is what the photography industry uses for the most part.

## 7.2 Camera Model

Each component to be simulated in the virtual camera is represented by a set of parameters that are updated once per frame, with a few parameters being defined as functions that vary based on an input variable.

The abstracted camera components and their parameters are listed below.

- Lens
  - Effective focal length
  - Focal distance
  - Image conjugate distance
  - Spherical aberration factor
  - Sagittal blur (as a function of off-axis distance)
  - Tangential blur (as a function of off-axis distance)
  - Distortion (as a function of off-axis distance)
  - Vignetting (as a function of off-axis distance)
- Aperture
  - F-number
  - Aperture shape (texture)
- Shutter
  - Exposure time
- Film
  - Film response curve (as a function of log exposure)

To represent physical lenses, information from their data sheets can be used to create text configuration files storing both their parameters and the acceptable range of values. The parameters that are functions of off-axis distance are listed as tables of distance vs parameter value. For easy access, the renderer converts these tables into one dimensional textures that shaders can sample at the appropriate coordinates to get the corresponding value.

### 7.2.1 Simplifications

Some simplifications are made in this model for ease of implementation. The entrance and exit pupils are assumed to be at the same position as the aperture stop, and their diameters are assumed to be equal to the aperture diameter. This allows the aperture stop to substitute for either pupil, and eliminates pupil magnification from many of the optical equations. For ease of control, the relative aperture it is also assumed that the effective aperture remains constant regardless of where the lens is focused.

## 7.3 Postprocessing Passes

When drawing, the renderer does an initial pass to light the scene. High dynamic range lighting values are used, storing the values in a floating point render target to hold the values with more precision. During this pass, per-pixel velocities are also calculated as the distance each vertex has traveled in texture space since the last frame, and rendered to a separate render target.

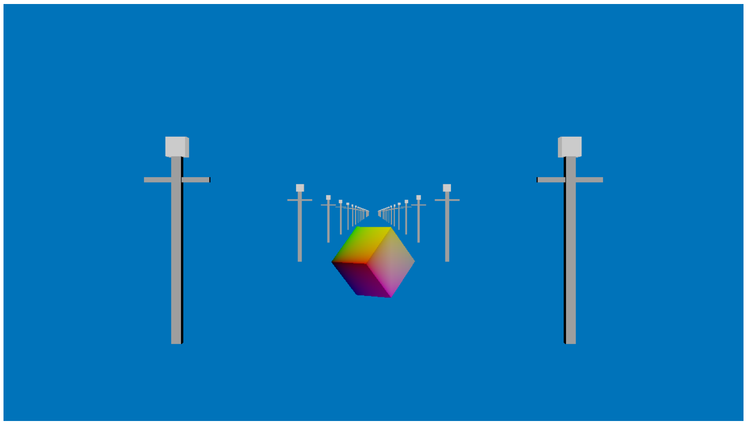

Figure 7.1: Initial pass

This is followed by a depth and bokeh prepass, which reads the depth buffer from the initial pass and writes out bokeh size and linear depth into a render target.

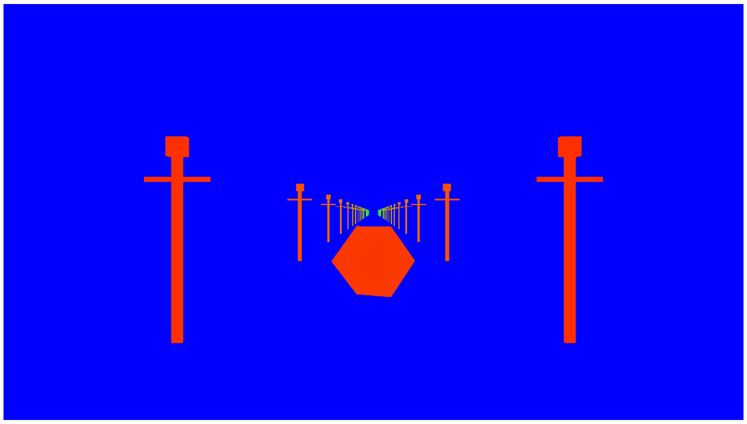

Figure 7.2: Depth map

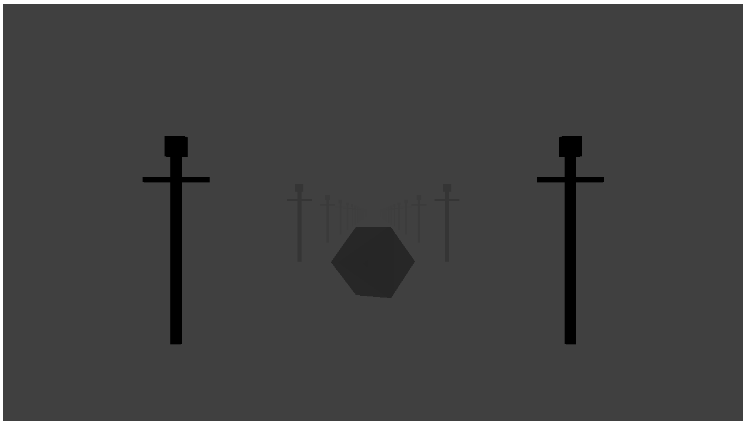

Figure 7.3: Bokeh size

Next, the velocity map is read and any motion blur is applied. For still photos, the amount of motion blur in the output would increase proportionally with object motion and exposure time. However, since motion information is only stored for the previous frame, the amount of motion blur is limited by the framerate at which the simulation is running.

Next, bokeh and spherical aberration are rendered together as they are dependent on each other. To accurately scatter image points, each circle of confusion is rendered with a quad covering the appropriate image area. The size of the circle of confusion is already precomputed, so the only thing left to do is modulate the image of the aperture with the correct spherical point spread function. One performance concern is that the number of pixels being drawn will quadruple with a doubling of the bokeh radius, potentially resulting in a large performance hit when most of the image is defocused. Although a gather-as-scatter approach would be significantly faster, (see [McG+12] for an example of this technique being applied to motion blur), the quad-based approach is necessary to render the bokeh shape accurately rather than just approximating it with a circle.

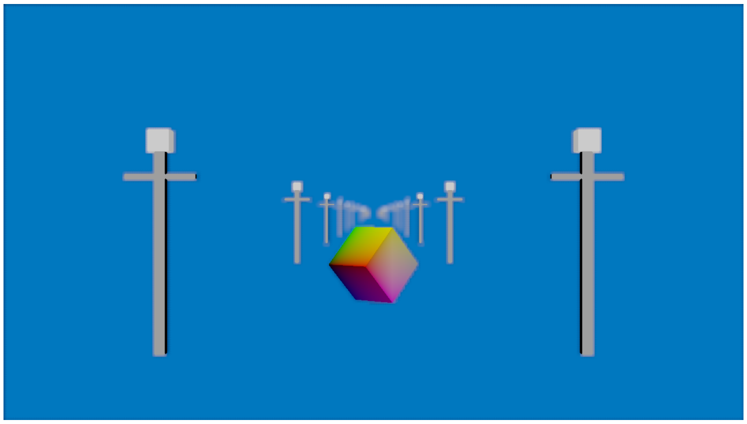

Figure 7.4: Bokeh blur

Lastly, the optical aberrations that vary with off-axis distance are applied. Astigmatism causes the image to blur in two orthogonal directions, so it has to be done in two separate passes. In the first aberration pass, distortion is applied, followed by astigmatic blur in the tangential direction. The second pass renders astigmatic blur in the sagittal direction, followed by applying a vignetting factor (see the Appendix 11 for shader code).

## 7.4 Long-Exposure Photography

Long-exposure photography is a photographic technique where the shutter is left open for an extended period of time. In such an image, stationary objects remain in sharp focus while moving objects blur into motion trails (which is particularly visible with light sources). This effect can be recreated quite efficiently with postprocessing, and with less work on the part of the photographer than with a real camera. For an actual long-exposure photograph, the photographer has to have a predetermined exposure duration, and adjust the illumination accordingly (see equation 4.1).

With a virtual camera model, the total exposure duration period can be broken down into individual frames, and accumulated into a final render target, allowing the exposure to be started and stopped arbitrarily. Image illumination need only be adjusted for a single frame, as applying proper blending during accumulation ensures that the final illumination will be the average, rather than the sum, of the illumination for the individual frames. This also works well with the basic motion blur algorithm that is currently implemented; although the renderer only captures motion blur for individual frames, the accumulation step helps make the blur paths appear connected rather than disjointed. For the accumulation steps, each successive frame needs to have a blend factor $a = 1/n$ where $n$ is the frame number since the long exposure began, and the render target should be have a blend factor of $1 - a$ applied to it.

## 7.5 User Interface

The program takes user input through keyboard presses, and displays relevant information to the user with a simple text-based UI. Major camera parameters can be adjusted directly through the keyboard, and although these values are adjusted in discrete steps, changes in parameter settings are also animated where they make sense. For example, when stepping the aperture size up/down or moving the focal distance forward/back, the current value is interpolated over time towards the new value, making transitions smoother. To simulate specific lenses and their aberrations, different lens preset files can be loaded as mentioned earlier in this chapter.

## 7.6 Omissions

There are a few effects that are omitted from the current camera model. The current vignetting model only accounts for illumination loss with distance from the axis; the change in bokeh shape due to optical vignetting is not included as simulating it accurately would require knowledge of the positions of each of the elements in the lens. It would be feasible to simulate the approximate result of the effect by performing the projection mentioned in section 4.4 and substituting the standard bokeh texture with this projected image. The additional parameters required would be the positions and images of the front lens element, entrance pupil, and rear lens element as a function of focal length and focal distance.

From the aberrations, coma, field curvature, and chromatic aberration are omitted due to time constraints and the effects they would have on the other optical parameters. Field curvature would be the easiest of these to implement, as the only modification required would be an adjustment to equation 5.3 to add an offset to the position of the film plane as a function of off-axis distance. Chromatic aberration would require modifications to both distortion and bokeh/spherical aberration to be implemented. Coma would probably be hardest to implement, as the resulting image will have to be rendered to a texture to be used in place of the bokeh texture, rotated to align in the correct direction, and scaled based on the off-axis distance.

# Chapter 8. Results

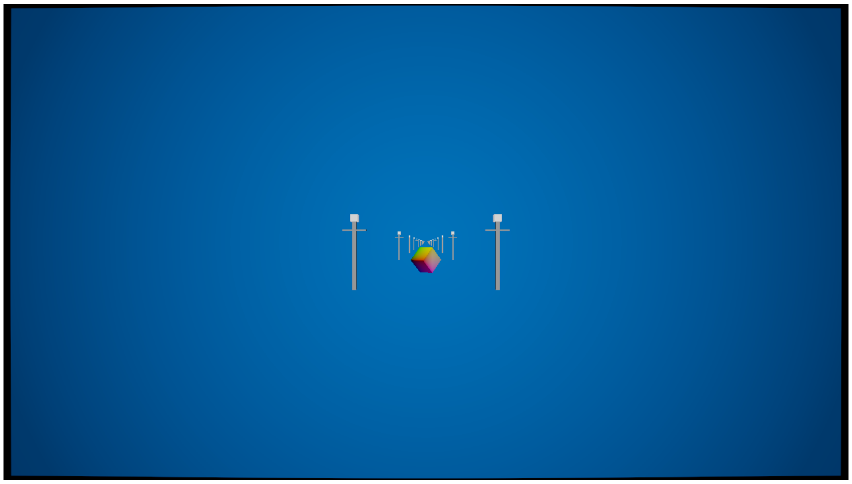

Figure 8.1: Distagon f/2.8 15mm

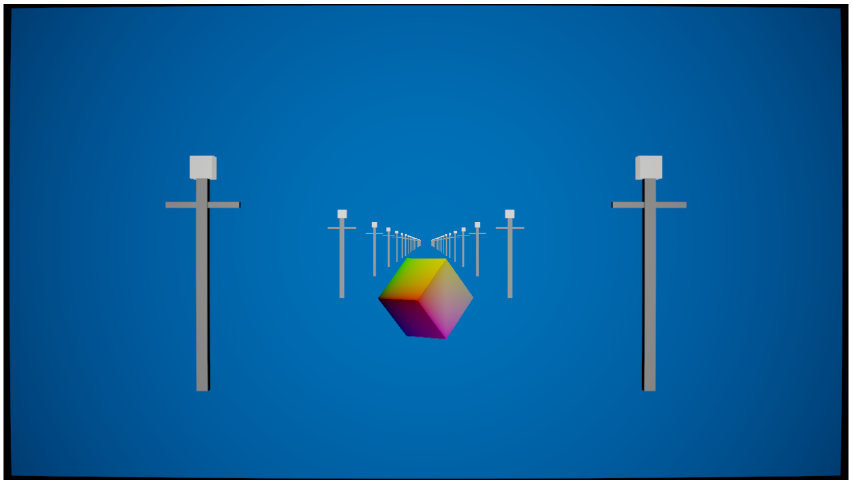

Figure 8.2: Planart f/1.4 50mm

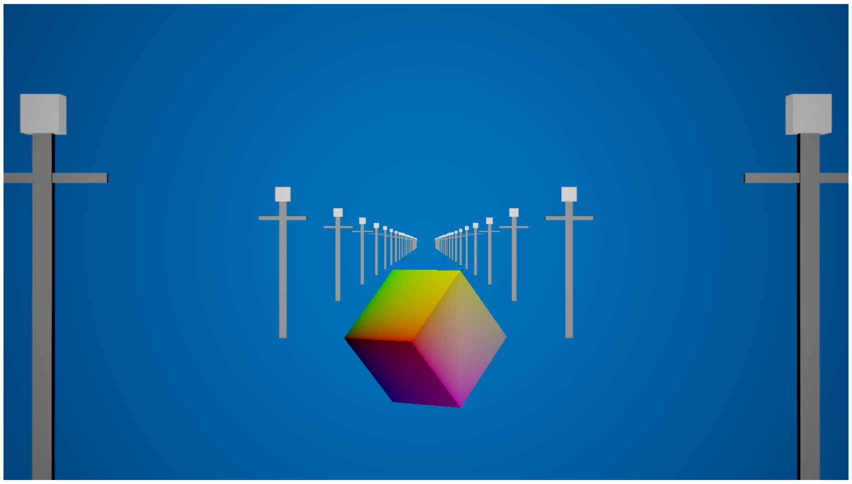

Figure 8.3: Planart f/1.4 85mm

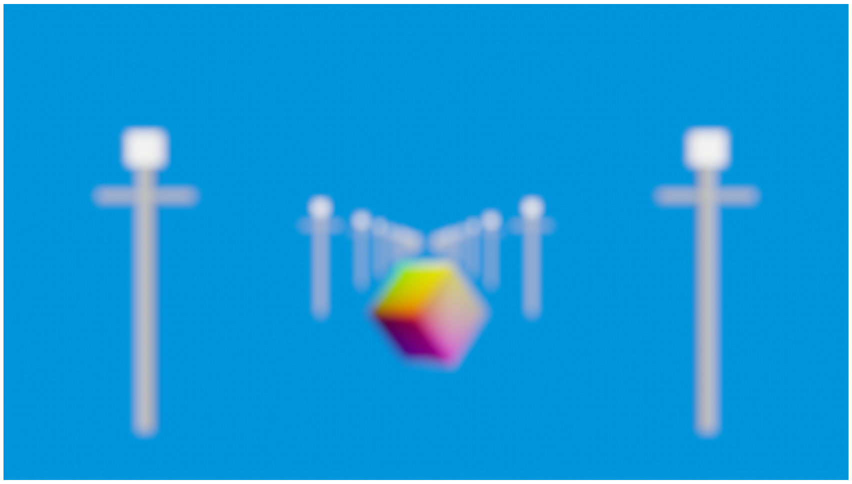

Figure 8.4: Bokeh/Defocus Blur

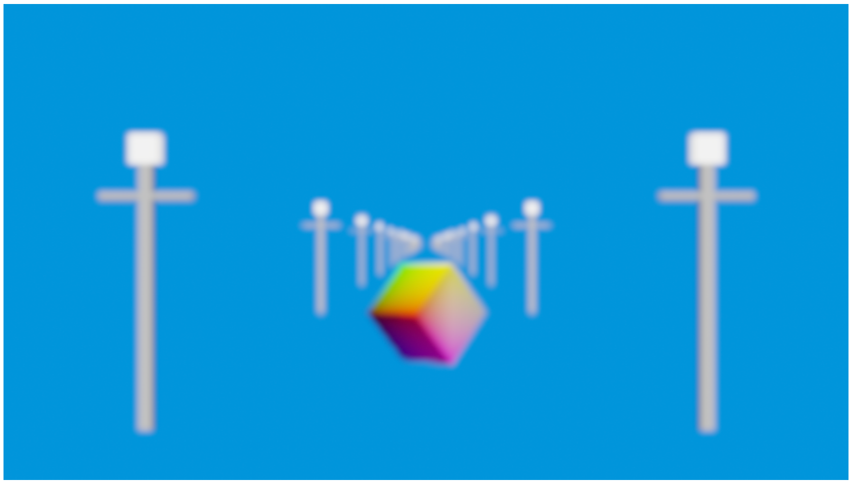

Figure 8.5: Bokeh with Positive Spherical Aberration

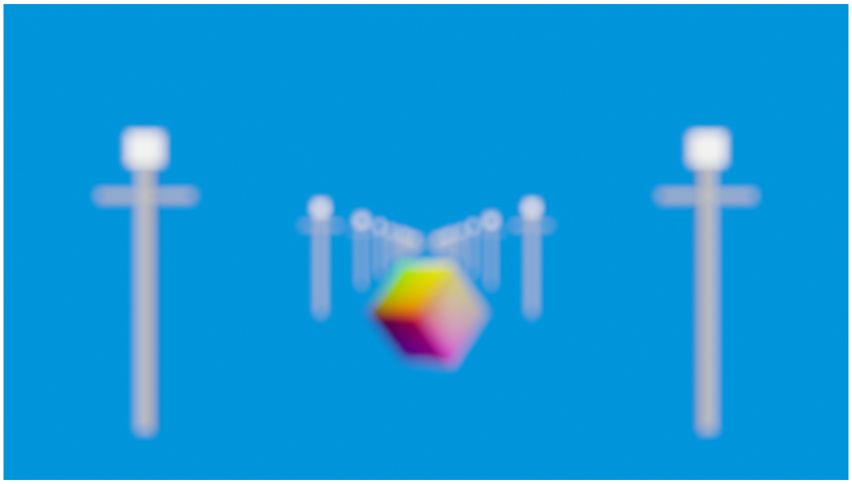

Figure 8.6: Bokeh with Negative Spherical Aberration

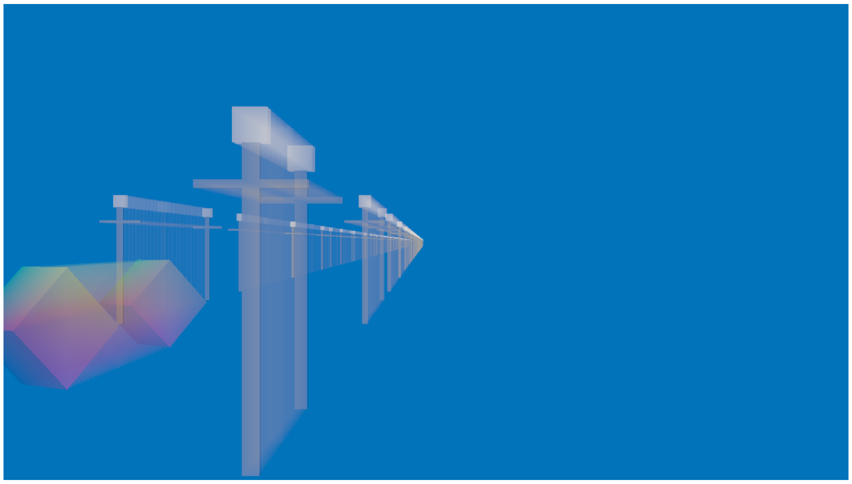

Figure 8.7: Long Exposure Motion Blur

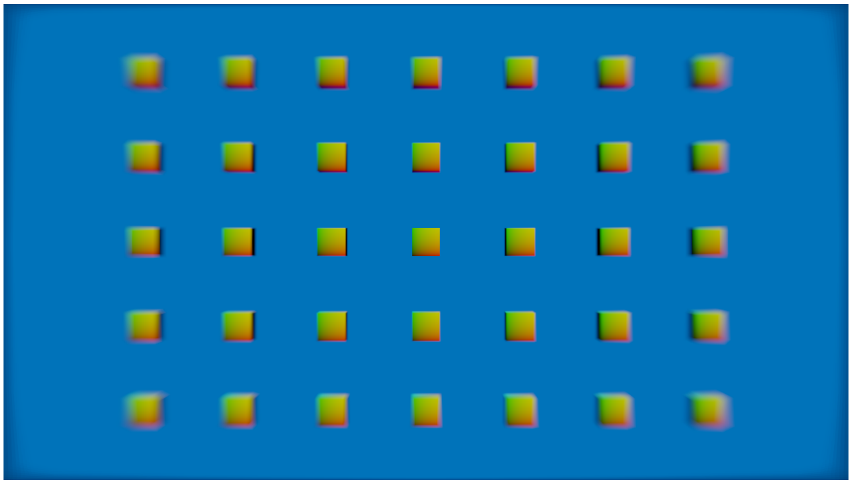

Figure 8.8: Astigmatism

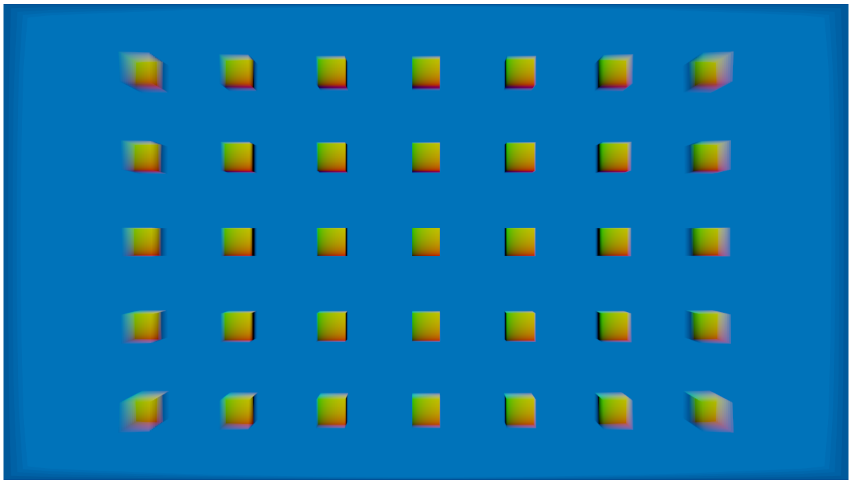

Figure 8.9: Sagittal Blur

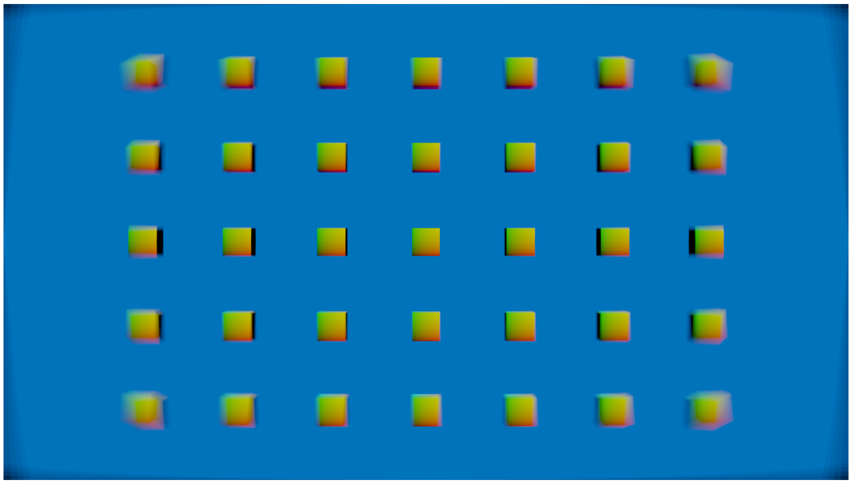

Figure 8.10: Tangential Blur

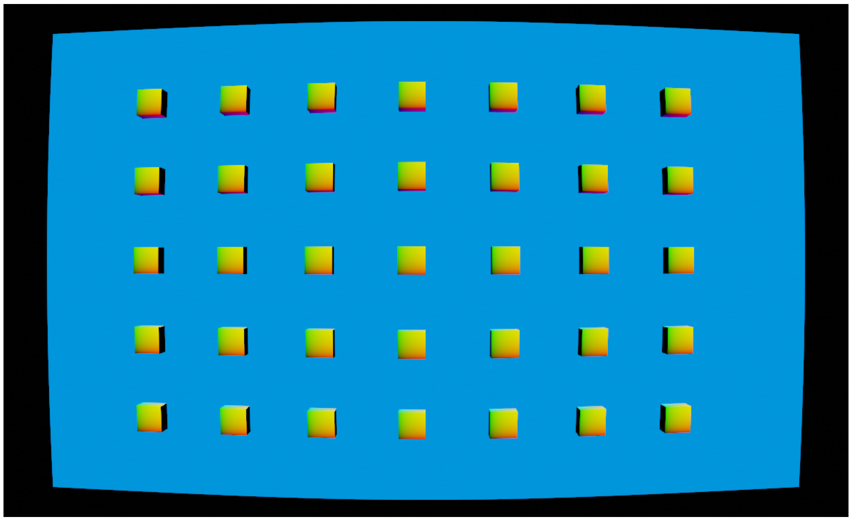

Figure 8.11: Barrel Distortion

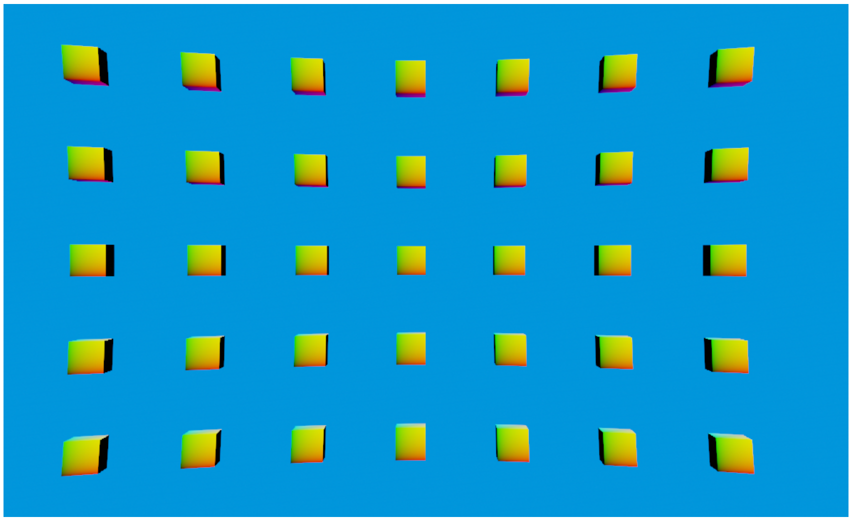

Figure 8.12: Pincushion Distortion

# Chapter 9. Performance Analysis

For performance tests, the renderer was run at a resolution of 1280x720 with the integrated GPU (Intel HD 4000) on an Intel Core i7-3635QM (2.40GHz x 4 cores). The quad-based approach to rendering bokeh and spherical aberration had the biggest performance on impact. Due to the massive amount of overdraw being performed with the technique, the fillrate on the GPU was easily saturated when most of the scene was defocused, and framerates hovered around 10 to 15 frames per second. On the other hand, the rest of the effects had little to no impact on performance, with astigmatic blur resulting in a steady 30fps in the worst case, and none of the other effects causing the framerate to drop below 60fps.

As blurring is being applied in addition to the quad-based bokeh rendering, memory bandwidth and fillrate would be the limiting factor for performance on most GPUs. A discrete GPU would probably be capable of achieving significantly higher framerates. Some additional optimizations that could be implemented to reduce memory usage would include

- Performing tone mapping earlier in the post processing chain to compress HDR values back into the standard dynamic range. This would allow subsequent render targets to use lower bit depths, reducing memory usage.

- Having an additional render target of half the size for bokeh rendering. Quads larger than a certain radius threshold would be rendered to the smaller render target, and this would be blended back together with the original image in the bokeh merge pass. This would reduce the number of pixels that are being drawn to, at the cost of losing some fine details on larger bokeh images.

# Chapter 10. Conclusion

Until now, there has not been a comprehensive camera model for real-time graphics that tried to simultaneously simulate defocus blur, motion blur, and third order lens aberrations. The goal of this camera model was to fill in that gap, and in doing so we have managed to define:

- A condensed set of equations that are directly applicable to virtual camera models
- A parameter-based model for describing and reproducing third order lens aberrations
- A flexible long-exposure model that allows the exposure to be started and stopped arbitrarily while allowing for easy control over the final exposure

The model is also performant enough to run most effects in real-time on an integrated laptop graphics card, so the average desktop gaming GPU should be able to handle these well.

## 10.1 Future Work

Most of the features that could be added to the camera model in this paper have already been covered. The entrance and exit pupils' positions and sizes could be simulated, and the effective aperture could be used in calculations instead of the relative aperture. The omitted effects of optical vignetting, coma, field curvature, and chromatic aberration could also be introduced to the model.

*Camera movements* are a less commonly discussed photographic technique that could probably be integrated into this camera model. A perspective control lens allows the photographer to control focus and convergence by adjusting the positioning and angle of the lens, and is commonly used to reduce geometric distortion due to perspective, or bring a larger portion of the scene into focus[Ray02].

Lens flare has not been considered yet in this model as its appearance is affected by a number of factors such as the number and positioning of lens elements and the coatings on the lenses, but recent research in the area has also been promising[Hul+11].

# Chapter 11. Appendix - Shader code

Data source for the camera presets: `http://lenses.zeiss.com/camera-lenses/carl-zeiss-camera-lenses/camera_lenses.html`

## 11.1 Circle of confusion diameter calculation

```
float CalculateCoCRadiusFilmS(float depthViewS) {
    // CoC parameters
    float S = depthViewS - NearClip; // Distance to object
    float u = FocalDistance;
    float v = LensDistance;
    float c1 = v * v * abs(S - u);
    float c2 = u * S * FNumber;
    float c = c1 / c2; // CoC diameter

    if(c < kMinCoCDiameter) {
        // Threshold to 0 if the diameter is small enough
        c = 0;
    } else if(c > kCoCSearchRadius * 2) {
        // Limit the CoC size to reduce overdraw
        c = kCoCSearchRadius * 2;
    }

    // Convert to radius and return
    return c * 0.5;
}
```

## 11.2 Depth of Field Geometry Shader

```
[maxvertexcount(16)]
void main(point GSInput center[1],
    inout TriangleStream<GSOutput> triStream)
{
    float2 depthAndCoCRadius = DepthAndCoCRadiusTex.Sample
        (LinearSampler, center[0].TexCoords);
    float linearDepth = depthAndCoCRadius.r;

    // Convert CoC radius from film space to image space
    float CoCRadiusFilmS = depthAndCoCRadius.g;
    float CoCDiameterPixelS = (2 * CoCRadiusFilmS)
        * ImageWidth / FilmWidth;

    // Only render if the CoC is wide enough,
    // and clamp if the CoC is too big
    if(CoCDiameterPixelS < kMinCoCPixels) {
        return;
    } else if(CoCDiameterPixelS > kCoCDiameterPixels) {
        CoCDiameterPixelS = kCoCDiameterPixels;
    }

    // Convert clamped diameter to clip space radius.
    float CoCXClipS = CoCDiameterPixelS / ImageWidth;
    float2 CoCRadiusClipS = float2(CoCXClipS,
        CoCXClipS * FilmWidth / FilmHeight);

    // The quad's alpha is proportional to the inverse of
    // the radius or diameter squared.
    float alphaModifier = 1 / (CoCDiameterPixelS
        * CoCDiameterPixelS);

    float4 positionClipS = float4(center[0].PositionClipS.xy,
        0, 1);
    float4 color = RenderTargetTex.Sample(LinearSampler,
        center[0].TexCoords);
    color.a *= alphaModifier;

    // Store the amount of SA to apply.
    float sa = clamp(SphericalAberrationFactor *
        CoCDiameterPixelS / kCoCDiameterPixels, -1, 1);

    // Modify Z to indicate whether the point is in the
    // foreground or background.
    if(linearDepth < FocalDistance) {
        // Foreground blur
        positionClipS.z = 0;

        // Negate the SA value to change the SA blur type
        sa *= -1;
    } else {
        // Background blur
        positionClipS.z = 1;
    }

    GSOutput v = (GSOutput)0;
    v.Color = color;
    v.SphericalAmount = sa;

    // Expand the point into a quad
    //-----------------------------------------------
    v.PositionClipS = positionClipS;

    // bottom left
    v.PositionClipS.xy = positionClipS.xy - CoCRadiusClipS;
    v.TexCoords = float2(0, 1);
    triStream.Append(v);

    // top left
    v.PositionClipS.x = positionClipS.x - CoCRadiusClipS.x;
    v.PositionClipS.y = positionClipS.y + CoCRadiusClipS.y;
    v.TexCoords = float2(0, 0);
    triStream.Append(v);

    // bottom right
    v.PositionClipS.x = positionClipS.x + CoCRadiusClipS.x;
    v.PositionClipS.y = positionClipS.y - CoCRadiusClipS.y;
    v.TexCoords = float2(1, 1);
    triStream.Append(v);

    // top right
    v.PositionClipS.xy = positionClipS.xy + CoCRadiusClipS;
    v.TexCoords = float2(1, 0);
    triStream.Append(v);

    triStream.RestartStrip();
}
```

## 11.3 Depth of Field Pixel Shader

```
PSOutput main(PSInput input)
{
    PSOutput output = (PSOutput)0;

    float4 color = input.Color;

    // Apply the bokeh mask
    color.a *= BokehTex.Sample(LinearSampler, input.TexCoords);

    // Z is set to 0 to indicate foreground blur,
    // and 1 to indicate background blur
    bool isForegroundBlur = input.PositionViewportS.z < 0.5;

    float positiveSA = PositiveSATex.Sample(LinearSampler,
        input.TexCoords);
    float zeroSA = ZeroSATex.Sample(LinearSampler,
        input.TexCoords);
    float negativeSA = NegativeSATex.Sample(LinearSampler,
        input.TexCoords);

    // Positive SA values indicate smooth blur,
    // and negative SA values indicate harsh blur.
    if(input.SphericalAmount > 0) {
        color.a *= lerp(zeroSA, positiveSA,
            input.SphericalAmount);
    } else {
        color.a *= lerp(zeroSA, negativeSA,
            -input.SphericalAmount);
    }

    if(isForegroundBlur) {
        // Foreground blur
        output.target0 = color;
    } else {
        // Background blur
        output.target1 = color;
    }

    return output;
}
```

## 11.4 Aberrations Pixel Shader - first pass

```
float4 main(PSInput input) : SV_TARGET
{
    // Convert from texture space to film space
    float x = (input.TexCoords.x - 0.5) * FilmWidth;
    float y = (input.TexCoords.y - 0.5) * FilmHeight;
    float radius = sqrt(x * x + y * y);

    // Coordinates for texture-based parameters
    // Convert from range of [0, 22] to [0, 1]
    float coord1D = radius / (22 * kMmToMeters);

    // Get the distortion amount
    float distortionAmt = 0.01
        * DistortionTex.Sample(LinearSampler, coord1D);
    float distortionFactor = 1 - distortionAmt;

    // Apply distortion
    float xNew = x * distortionFactor;
    float yNew = y * distortionFactor;
    radius *= distortionFactor;
    coord1D = radius / (22 * kMmToMeters);


    // Convert back to texture space to get sample point
    float u = xNew / FilmWidth + 0.5;
    float v = yNew / FilmHeight + 0.5;

    // Astigmatism - tangential blur
    float tangentialFactor = 0.01
        * TangentialTex.Sample(LinearSampler, coord1D);

    // Sample the source image.
    float2 curTexCoords = float2(u, v);
    float4 color = SourceTex.Sample(LinearSampler, curTexCoords);
    int sampleCount = 1;

    // Sample the color buffer along the blur vector.
    if(tangentialFactor != 0) {
        float tangentialDeltaFactor = tangentialFactor
            / (numSamples * 2);
        // Tangential vector in texture space
        float2 tangentialVector =
            float2(-y / FilmHeight, x / FilmWidth);
        for(int i = 1; i <= numSamples; ++i) {
            // Tangential samples
            float2 offset = i * tangentialDeltaFactor
                * tangentialVector;
            color += SourceTex.Sample(LinearSampler,
                curTexCoords + offset);
            color += SourceTex.Sample(LinearSampler,
                curTexCoords - offset);
        }
        sampleCount += numSamples * 2;
    }

    // Average all of the samples to get the final blur color.
    color /= sampleCount;

    return color;
}
```

## 11.5 Aberrations Pixel Shader - second pass

```
float4 main(PSInput input) : SV_TARGET
{
    // Convert from texture space to film space
    float x = (input.TexCoords.x - 0.5) * FilmWidth;
    float y = (input.TexCoords.y - 0.5) * FilmHeight;
    float radius = sqrt(x * x + y * y);

    // Coordinates for texture-based parameters
    // Convert from range of [0, 22] to [0, 1]
    float coord1D = radius / (22 * kMmToMeters);

    // Astigmatism - sagittal blur
    float sagittalFactor = 0.01
        * SagittalTex.Sample(LinearSampler, coord1D);

    // Sample the source image.
    float4 color = SourceTex.Sample(LinearSampler,
        input.TexCoords);
    int sampleCount = 1;

    // Sample the color buffer along the blur vector.
    if(sagittalFactor != 0) {
        float sagittalDeltaFactor = sagittalFactor
            / (numSamples * 2);
        // Sagittal vector in texture space
        float2 sagittalVector =
            float2(x / FilmWidth, y / FilmHeight);
        for(int i = 1; i <= numSamples; ++i)
        {
            // Sagittal samples
            float2 offset = i * sagittalDeltaFactor
                * sagittalVector;
            color += SourceTex.Sample(LinearSampler,
                input.TexCoords + offset);
            color += SourceTex.Sample(LinearSampler,
                input.TexCoords - offset);
        }
        sampleCount += numSamples * 2;
    }

    // Average all of the samples to get the final blur color.
    color /= sampleCount;

    // Vignetting
    float vignetting = 0.01
        * VignettingTex.Sample(LinearSampler, coord1D);
    float4 finalColor = color * vignetting;

    return finalColor;
}
```

# Bibliography

[AHA11] Magnus Andersson, Jon Hasselgren, and Tomas Akenine-Möller. "Depth buffer compression for stochastic motion blur rasterization". In: *Proceedings of the ACM SIGGRAPH Symposium on High Performance Graphics*. HPG '11. Vancouver, British Columbia, Canada: ACM, 2011, pp. 127–134. ISBN: 978-1-4503-0896-0. DOI: 10.1145/2018323.2018343. URL: http://doi.acm.org/10.1145/2018323.2018343.

[Dem04] Joe Demers. *GPU Gems - Depth of Field: A Survey of Techniques*. 2004. URL: http://developer.nvidia.com/GPUGems/gpugems_ch23.html.

[Hab10] John Hable. *Why a Filmic Curve Saturates Your Blacks*. 2010. URL: http://filmicgames.com/archives/190.

[Hul+11] Matthias Hullin, Elmar Eisemann, Hans-Peter Seidel, and Sungkil Lee. “Physically-based real-time lens flare rendering”. In: *ACM SIGGRAPH 2011 papers*. SIGGRAPH ’11. Vancouver, British Columbia, Canada: ACM, 2011, 108:1–108:10. ISBN: 978-1-4503-0943-1. DOI: 10.1145/1964921.1965003. URL: http://doi.acm.org/10.1145/1964921.1965003.

[KMH95] Craig Kolb, Don Mitchell, and Pat Hanrahan. “A realistic camera model for computer graphics”. In: *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*. SIGGRAPH ’95. New York, NY, USA: ACM, 1995, pp. 317–324. ISBN: 0-89791-701-4. DOI: 10.1145/218380.218463. URL: http://doi.acm.org/10.1145/218380.218463.

[LB11] Michael Langford and Efthimia Bilissi. *Langford’s Advanced Photography*. Focal Press, 2011.

[LES10] Sungkil Lee, Elmar Eisemann, and Hans-Peter Seidel. “Real-time lens blur effects and focus control”. In: *ACM SIGGRAPH 2010 papers*. SIGGRAPH ’10. Los Angeles, California: ACM, 2010, 65:1–65:7. ISBN: 978-1-4503-0210-4. DOI: 10.1145/1833349.1778802. URL: http://doi.acm.org/10.1145/1833349.1778802.

[LFS10] Michael Langford, Anna Fox, and Richard Sawdon Smith. *Langford’s Basic Photography*. Focal Press, 2010.

[McG+10] Morgan McGuire, Eric Enderton, Peter Shirley, and David Luebke. “Real-Time Stochastic Rasterization on Conventional GPU Architectures”. In: *Proceedings of High Performance Graphics 2010*. Saarbrucken, Germany, June 2010. URL: http://graphics.cs.williams.edu/papers/AOVHPG10.

[McG+12] Morgan McGuire, Padraic Hennessy, Michael Bukowski, and Brian Osman. “A reconstruction filter for plausible motion blur”. In: *Proceedings of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*. I3D ’12. Costa Mesa, California: ACM, 2012, pp. 135–142. ISBN: 978-1-4503-1194-6. DOI: 10.1145/2159616.2159639. URL: http://doi.acm.org/10.1145/2159616.2159639.

[Mun+11] Jacob Munkberg, Petrik Clarberg, Jon Hasselgren, Robert Toth, Masamichi Sugihara, and Tomas Akenine-Möller. “Hierarchical stochastic motion blur rasterization”. In: *Proceedings of the ACM SIGGRAPH Symposium on High Performance Graphics*. HPG ’11. Vancouver, British Columbia, Canada: ACM, 2011, pp. 107–118. ISBN: 978-1-4503-0896-0. DOI: 10.1145/2018323.2018341. URL: http://doi.acm.org/10.1145/2018323.2018341.

[Nas10] H. H. Nasse. *Depth of Field and Bokeh*. 2010. URL: http://www.zeiss.com/C12567A8003B8B6F/EmbedTitelIntern/CLN_35_Bokeh_EN/$File/CLN35_Bokeh_en.pdf.

[Ray02] Sidney F. Ray. *Applied Photographic Optics*. Focal Press, 2002.

[Sensito] *Basic Sensitometry and Characteristics of Film*. URL: http://motion.kodak.com/motion/uploadedFiles/US_plugins_acrobat_en_motion_newsletters_filmEss_06_Characteristics_of_Film.pdf.