Technische Raytracer

Lens Flare Simulation in PBRT v4

Inhaltsverzeichnis

- Technische Raytracer
  - Lens Flare Simulation in PBRT v4
- Inhaltsverzeichnis
  - 1 Einleitung
    - 1.1 Motivation und Hintergrund
    - 1.2 Zielsetzung
  - 2 Theoretische Grundlagen
  - 3 Vorgehensweise
    - 3.1 Evaluierung von Lösungsansätzen
    - 3.2 Szenen- / Versuchsaufbau
    - 3.3 Mess- und Bewertungsschema
  - 4 Implementierung
    - 4.1 Datenstrukturen
    - 4.2 Klassen- und Softwarearchitektur
    - Ablauf der Lens-Flare-Strahlverfolgung
    - Sequenzdiagramm
    - 4.3 Integrator-Anbindung
  - 4.4 Zylinderlinsen und Anamorphie
    - Intersection und Normalenberechnung
    - Fokusberechnung und Astigmatismus
      - Wahl des Referenzobjektivs
      - Praktische Umsetzung
    - 4.5 Kamera- und Szenenparameter für Lens-Flares
  - 5 Ergebnisse
    - 5.1 Messergebnisse
    - 5.2 Auswertung
  - 6 Diskussion
    - 6.1 Interpretation der Ergebnisse
    - 6.2 Fehlerbetrachtung
    - 6.3 Nachteil bei reflektierenden Lichtquellen
  - 7. Fazit und Ausblick
  - 8 Literaturverzeichnis
  - 9 Anhang
    - Eigenständigkeitserklärung

1 Einleitung

1.1 Motivation und Hintergrund

In der realen Fotografie und Kinematografie wird das Erscheinungsbild eines Bildes nicht nur durch die Szene selbst, sondern maßgeblich auch durch die Eigenschaften des verwendeten Kamerasystems bestimmt. Insbesondere optische Effekte, die direkt im Objektiv entstehen, tragen wesentlich zur visuellen Charakteristik einer Aufnahme bei.

Lens Flares gehören zu diesen optischen Effekten und entstehen durch Reflexionen an Glas-Luft-Grenzflächen innerhalb eines Objektivs. Ein Teil des einfallenden Lichts wird an den Linsenoberflächen reflektiert, mehrfach im Objektiv umgelenkt und kann schließlich als sogenannte Ghost-Artefakte auf der Bildebene sichtbar werden. Besonders bei sehr hellen Lichtquellen prägen diese Effekte die visuelle Wahrnehmung realer Kameraaufnahmen.

In vielen Rendering-Systemen werden Lens Flares als Post-Processing-Effekt realisiert. Dieser Ansatz ist effizient, berücksichtigt jedoch weder das konkrete Linsendesign noch den Einfluss von Blende, Fokus oder optischen Abbildungsfehlern. Eine Simulation der Effekte direkt im Linsensystem ist daher physikalisch plausibler und besser geeignet, realistische Kameraästhetik zu erzeugen.

Auch haben dabei anamorphotische Objektive eine Bedeutung, die zylindrische Linsenelemente mit richtungsabhängiger Krümmung verwenden. Diese führen zu anisotropen Lens Flares, beispielsweise charakteristischen horizontalen Lichtstreifen. Solche Effekte lassen sich mit rein sphärischen Linsenmodellen nicht abbilden und motivieren eine Erweiterung des optischen Modells.

1.2 Zielsetzung

Ziel dieser Arbeit ist es, den physikalisch basierten Renderer PBRT v4 so zu erweitern, dass Lens Flares als Folge interner Reflexionen im Linsensystem simuliert werden können. Die Umsetzung soll es ermöglichen, Lens Flares isoliert vor schwarzem Hintergrund darzustellen, um deren Form, Lage und Intensität gezielt zu analysieren.

Dabei werden ausschließlich Lichtpfade mit maximal zwei internen Reflexionen berücksichtigt, da diese den wesentlichen sichtbaren Beitrag zu Lens Flares liefern und gleichzeitig eine praktikable rechnerische Komplexität gewährleisten.

Darüber hinaus ist es Ziel der Arbeit, das Kameramodell konzeptionell auf die Unterstützung von Zylinderlinsen vorzubereiten, um anisotrope und anamorphotische Lens-Flare-Effekte untersuchen zu können.

2 Theoretische Grundlagen

Die Simulation von Lens Flares basiert auf grundlegenden Modellen der geometrischen und physikalischen Optik. Beim Übergang von Licht zwischen zwei Medien mit unterschiedlichem Brechungsindex wird der

größte Teil des Lichtes gebrochen, während stets ein kleiner Anteil reflektiert wird.

Fresnel-Gleichungen
Der reflektierte Lichtanteil ist vom Einfallswinkel θ_i, dem Transmissionswinkel θ_t sowie den Brechungsindizes der beteiligten Medien (n_1, n_2) abhängig. Für unpolarisiertes Licht berechnet sich der Reflexionsgrad F_r als Mittelwert der s- und p-polarisierten Anteile:

$$R_s = \left| \frac{n_1 \cos\theta_i - n_2 \cos\theta_t}{n_1 \cos\theta_i + n_2 \cos\theta_t} \right|^2, \quad R_p = \left| \frac{n_1 \cos\theta_t - n_2 \cos\theta_i}{n_1 \cos\theta_t + n_2 \cos\theta_i} \right|^2$$

$$F_r = \frac{R_s + R_p}{2}$$

Ghosting und Apertur
Lens Flares entstehen hauptsächlich durch mehrfache interne Reflexionen zwischen Linsenoberflächen. Insbesondere Lichtpfade mit zwei Reflexionen tragen wesentlich zu sichtbaren Ghost-Artefakten bei. Reflexionsordnungen höherer Ordnung liefern aufgrund der Potenzierung des Faktors F_r (z.B. $F_r^4 \ll F_r^2$) in der Regel nur einen geringen visuellen Beitrag und werden vernachlässigt.

Fokus und Astigmatismus
Für die Fokusberechnung verwendet PBRT die Thick-Lens-Approximation. Dieses Modell setzt Rotationssymmetrie voraus und ist daher für zylindrische Linsen nur eingeschränkt geeignet. Zylinderlinsen führen zu Astigmatismus, wodurch Licht in horizontaler und vertikaler Richtung unterschiedlich fokussiert wird, was anisotrope Lens-Flare-Strukturen (Streaks) verursacht.

3 Vorgehensweise

3.1 Evaluierung von Lösungsansätzen

Die Entwicklung der Lens-Flare-Simulation erfolgte iterativ. Dabei wurde zunächst ein naiver stochastischer Ansatz verfolgt, der sich als ineffizient erwies, welcher anschließend durch einen deterministischen, kombinatorischen Ansatz ersetzt wurde.

Ansatz 1: Stochastisches Raytracing

Der initiale Lösungsansatz zielte darauf ab, die Flares direkt innerhalb des bestehenden Path-Tracing-Prozesses von PBRT v4 zu generieren. Die Klasse `RealisticCamera` implementiert bereits die Brechung (Refraktion) von Strahlen durch das Linsensystem mittels einer gerichteten `for`-Schleife über die Linsenelemente. Die Idee bestand darin, diese deterministische Schleife in eine probabilistische `while`-Schleife umzuwandeln. An jeder Linsenoberfläche i sollte basierend auf dem Fresnel-Reflexionsgrad F_r entschieden werden, ob ein Strahl gebrochen oder reflektiert wird.

Hierzu wird eine Zufallsvariable $\xi \in [0, 1)$ gezogen:

- Wenn $\xi < F_r$: Der Strahl wird reflektiert (Richtungsumkehr im Linsensystem).
- Wenn $\xi \geq F_r$: Der Strahl wird gebrochen (Standardverhalten).

Problem:
Lens Flares entstehen primär durch Pfade mit zwei internen Reflexionen. Da der Reflexionsgrad F_r an den Linsen sehr gering ist (typischerweise $F_r < 0.04$ oder 4%), ist die Wahrscheinlichkeit P_{path} für einen vollständigen Ghost-Pfad extrem niedrig.

Bei zwei Reflexionen ergibt sich eine Wahrscheinlichkeit im Bereich von $0.04 \cdot 0.04 = 0.0016$ (0.16%). Ein Monte-Carlo-Integrator benötigt unzählige Samples, um solche seltenen Pfade zu finden. Das Ergebnis war ein Bild, das fast ausschließlich aus Rauschen bestand oder komplett schwarz blieb, da kaum ein Strahl zufällig die korrekte Sequenz (Licht $\rightarrow$ Linse $\rightarrow$ Reflexion $\rightarrow$ Reflexion $\rightarrow$ Sensor) fand.

Abbildung 1: Stochastisches Raytracing

Ansatz 2: Explizite Kombinatorik (Finaler Ansatz)

Um das Rauschproblem zu lösen, wurde das probabilistische Sampling-Problem in ein kombinatorisches Enumerations-Problem überführt, basierend auf der Methode von *Hullin et al. (2011)*. Anstatt darauf zu warten, dass ein Strahl zufällig reflektiert wird, werden alle möglichen Reflexionspaare explizit generiert. Der Algorithmus zwingt den Strahl deterministisch, an zwei spezifischen Flächen zu reflektieren. Dadurch wird garantiert, dass jeder berechnete Strahl zum gewünschten Ghost-Effekt beiträgt, sofern er nicht blockiert wird.

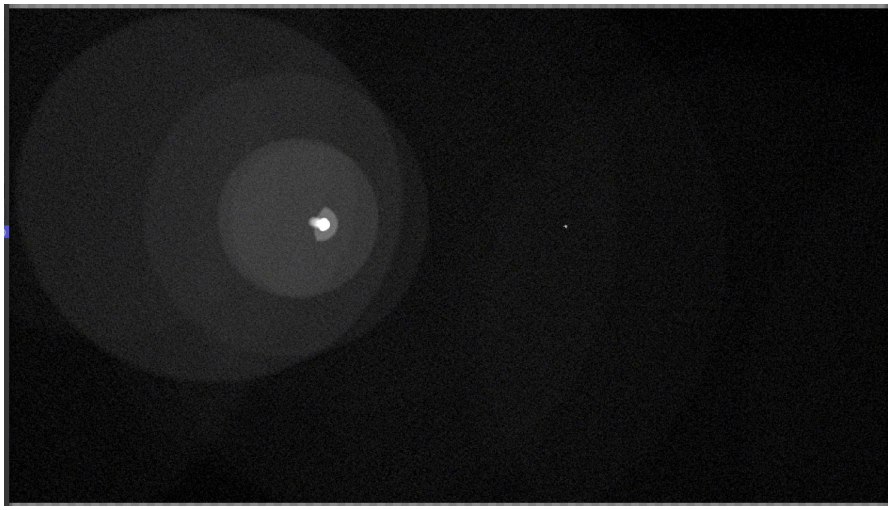


Abbildung 2: Explizite Kombinatorik

3.2 Szenen- / Versuchsaufbau

Für die Untersuchung der Lens-Flare-Effekte wurde eine stark vereinfachte, synthetische Szene verwendet. Die Szene wird mit der `realisticCamera` Klasse gerendert, die ein physikalisch beschriebenes 50mm double gauss Linsenmodell nutzt. Der Blendendurchmesser ist mit 3 mm neutral gewählt, um das Auftreten von Lens-Flares nicht zu behindern und diese dennoch Scharf zu halten.

Als Lichtquelle dient ein extrem helles, diffuses Flächenlicht, das in großer Entfernung zur Kamera positioniert ist und durch eine kleine Kugel näherungsweise punktförmig realisiert wird. Weitere Geometrien oder Lichtquellen sind nicht vorhanden, wodurch ein vollständig schwarzer Hintergrund entsteht.

3.3 Mess- und Bewertungsschema

Da sich Lens-Flares als Folge komplexer, mehrfacher interner Reflexionen im Linsensystem nur eingeschränkt analytisch beschreiben lassen, wurde in dieser Arbeit auf eine rein formelbasierte Validierung verzichtet. Stattdessen erfolgte die Bewertung der Ergebnisse primär qualitativ-visuell anhand von Referenzbeobachtungen aus der realen Fotografie und bekannten Darstellungen aus der Literatur.

Die erzeugten Lens-Flares wurden hinsichtlich ihrer

- räumlichen Lage
- Form
- spektralen Eigenschaften (chromatische Aberration etc.)

analysiert und mit typischen Erscheinungsformen realer Objektive verglichen. Dabei zeigte sich, dass die simulierten Effekte in Struktur und Verhalten mit den erwarteten optischen Phänomenen gut übereinstimmen.

Ergänzend wurde versucht, die Resultate mit einer professionellen Optiksimulationssoftware zu vergleichen. Hierzu kam das Programm `Zemax OpticStudio` zum Einsatz. In der verwendeten Version stand jedoch die `Stray Light Analysis / Ghost-Tracing` Funktionalität nicht zur Verfügung, sodass ein direkter Vergleich der simulierten Lens-Flares mit einer optischen Referenzsimulation nicht möglich war.

4 Implementierung

4.1 Datenstrukturen

Die bestehende Linsenbeschreibung von PBRT bildet die Grundlage der Erweiterung.

- **LensElementInterface:** Diese Struktur repräsentiert eine einzelne Grenzfläche. Sie wurde um `curvatureRadiusX` und `curvatureRadiusY` für anamorphotische Linsen erweitert. Zusätzlich speichert die Struktur die Sellmeier-Koeffizienten ($B_{1..3}, C_{1..3}$) zur Simulation der chromatischen Aberration (Dispersion). Der wellenlängenabhängige Brechungsindex $\eta(\lambda)$ berechnet sich zur Laufzeit gemäß der Sellmeier-Gleichung:

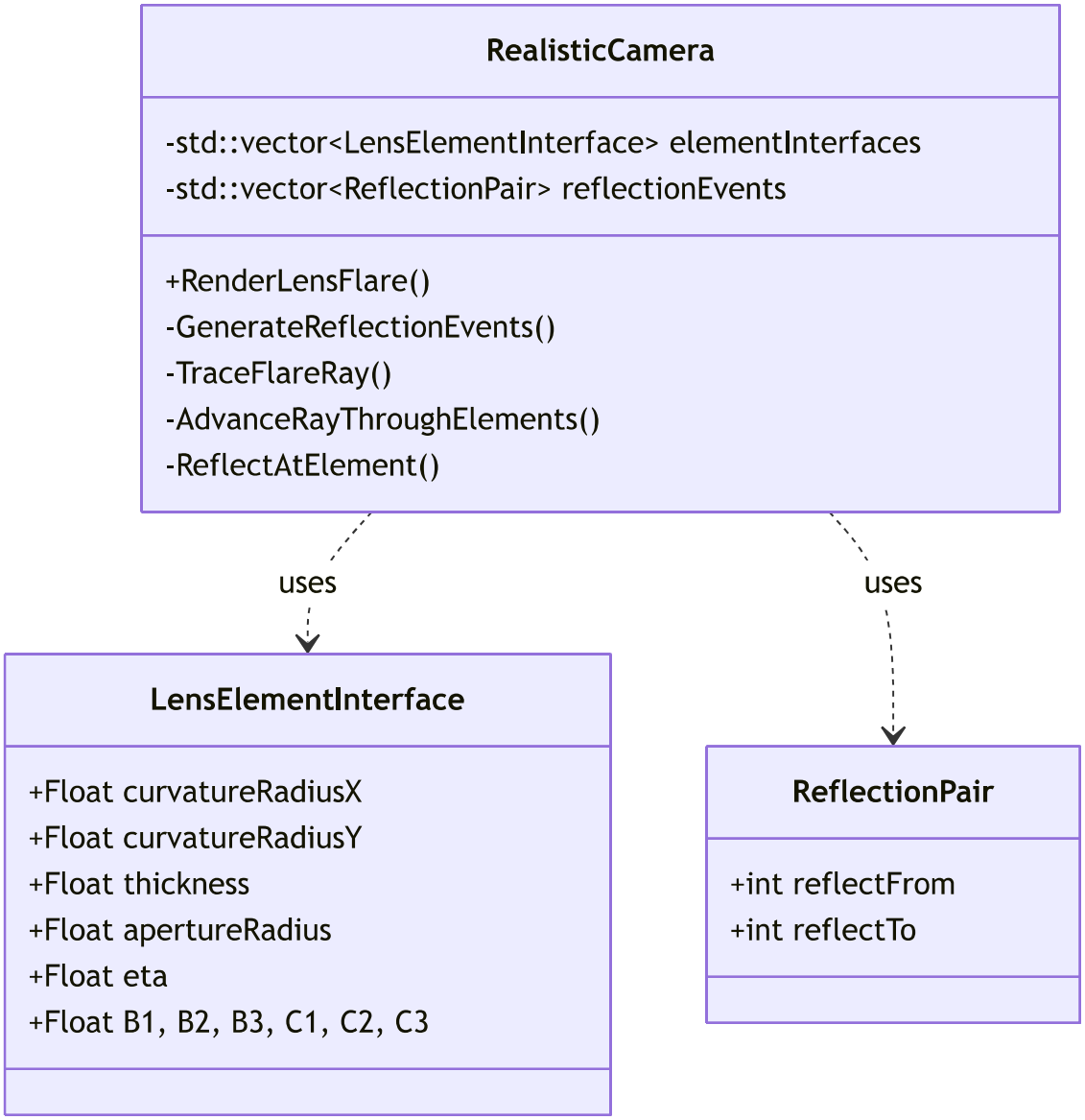
$$n^2(\lambda) \;=\; 1 + \frac{B_1\lambda^2}{\lambda^2 - C_1} + \frac{B_2\lambda^2}{\lambda^2 - C_2} + \frac{B_3\lambda^2}{\lambda^2 - C_3}$$

Wobei λ die Wellenlänge in Mikrometern ist.

- **ReflectionPair:** Ein `ReflectionPair` beschreibt ein geordnetes Paar zweier Linsenflächen (`reflectFrom`, `reflectTo`). Diese Paare repräsentieren Reflexionspfade zweiter Ordnung.

4.2 Klassen- und Softwarearchitektur

Die Implementierung gliedert sich nahtlos in die bestehende Objektstruktur ein. Die Klasse `RealisticCamera` wird um spezifische Methoden zur Flare-Berechnung erweitert.



Ablauf der Lens-Flare-Strahlverfolgung

1. **Sampling der Lichtquelle**
    - Position und Richtung des Strahls werden über Low-Discrepancy-Sequenzen (`RadicalInverse`) auf der Lichtquelle gesampelt.

- Wellenlänge(n) des Strahls werden entweder monchromatisch oder als sichtbares Spektrum ausgewählt (`SampleVisibleWavelengths`).
- Die Lichtintensität wird aus `Light::SampleLe()` entnommen.

2. **Auswahl eines Reflexionspfads**

   - Ein `ReflectionPair` wird aus der Liste aller möglichen Linsenreflexionskombinationen (`reflectionEvents`) ausgewählt.

3. **Sampling des Strahlstarts auf der ersten Linsenfläche**

   - Ein Punkt auf der ersten Linse wird über `SampleUniformDiskConcentric` innerhalb der Apertur ausgewählt.
   - Der Strahl wird vom Lichtpunkt zu diesem Linsenpunkt ausgerichtet.

4. **Strahlverfolgung durch das Linsensystem**

   - Refraktion durch Linsen: `AdvanceRayThroughElements()`
   - Reflexion an einer Fläche: `ReflectAtElement()` (optional mit Sellmeier-Brechungsindex `Sellmeier()`)
   - Strahl wird durch die dazwischenliegenden Linsen refraktiert.
   - Reflexion an der zweiten Fläche.
   - Weiter durch die restlichen Linsen refraktiert bis zur Filmebene.

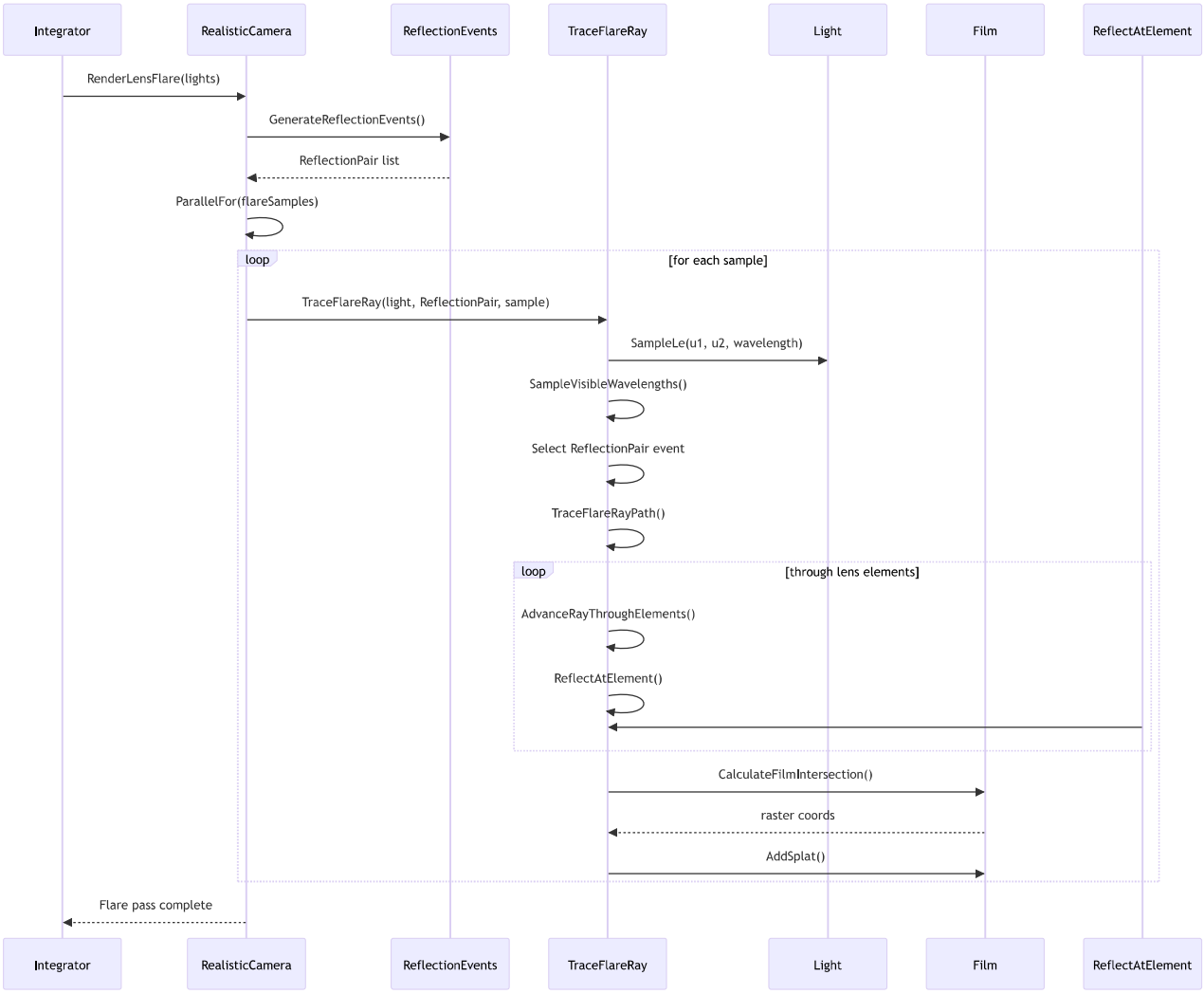
5. **Projektion auf den Film und Rasterisierung**

   - Schnittpunkt mit der Filmebene berechnen.
   - Überprüfung, ob der Punkt innerhalb der Filmgrenzen liegt.
   - Umrechnung in Rasterkoordinaten (`physicalExtent.Offset()`).

6. **Akkumulation der Flare-Beiträge**

   - Intensität auf Pixel anwenden (`Film::AddSplat()`).
   - Bei chromatischer Dispersion wird jeder Wellenlängenkanal separat hinzugefügt.
   - Normalisierung durch die Gesamtanzahl der Flare-Samples.

Sequenzdiagramm



Generierung der Reflexionsereignisse In der Funktion `GenerateReflectionEvents()` werden alle möglichen Kombinationen von zwei reflektierenden Linsenflächen erzeugt. Gemäß Hullin et al. (2011) berechnet sich die Anzahl N_{ghosts} der möglichen Pfade 2. Ordnung für ein System mit n Elementen wie folgt:

$$N_{ghosts} = \frac{n(n-1)}{2}$$

Hierbei werden Aperturblenden explizit ausgeschlossen, da sie absorbierend wirken ($R_{aperture} = 0$).

Bei der von uns verwendeten Double-Gauss-Linse ergeben sich aus 6 Linsenelementen also 15 mögliche Kombinationen:

$$N_{ghosts} = \frac{6 \cdot (6-1)}{2} = 15$$

Strahlverfolgung und Physik Die zentrale Funktion `TraceFlareRay()` steuert den Prozess.

1. `AdvanceRayThroughElement()`: Propagiert den Strahl mittels Brechung (Snellius) durch ein Element.
2. `ReflectAtElement()`: Führt die Spiegelung durch. Hierbei wird die Intensität I des Strahls mit dem Fresnel-Term $F_r(\theta_i, \eta)$ gewichtet.

Trifft der Strahl nach der Sequenz *Licht* $\rightarrow$ *Brechung* $\rightarrow$ *Reflexion 1* $\rightarrow$ *Reflexion 2* $\rightarrow$ *Brechung* $\rightarrow$ *Film* auf den Sensor, wird sein Beitrag normalisiert und als Splat akkumuliert.

4.3 Integrator-Anbindung

Die Lichtquellen werden nicht innerhalb der Kamera definiert, sondern vom Integrator als Vektor aller aktiven Szenenlichter bereitgestellt. Dieser Vektor wird an den Lens-Flare-Pass der Kamera übergeben und dient dort ausschließlich als Ausgangspunkt für das Sampling von Emissionsstrahlen zur Simulation interner Reflexionen im Linsensystem.

Der bestehende Integrator von PBRT wurde hierfür um einen zusätzlichen Verarbeitungsschritt nach dem regulären Rendering erweitert. Nach Abschluss der Bildberechnung wird geprüft, ob die verwendete Kamera eine RealisticCamera ist. In diesem Fall wird die Funktion `RenderLensFlare()` aufgerufen, welche die Lens-Flare-Beiträge berechnet und additiv auf dem Film akkumuliert. Der eigentliche Integrationsalgorithmus zur Berechnung der globalen Beleuchtung bleibt dabei unverändert.

4.4 Zylinderlinsen und Anamorphie

Zylindrische beziehungsweise anamorphotische Linsen unterscheiden sich von sphärischen Linsen durch eine richtungsabhängige Krümmung und besitzen keine Rotationssymmetrie. Während sphärische Linsen durch identische Krümmungsradien beschrieben werden können, gilt für zylindrische Linsen: $R_x \neq R_y$

Diese Anisotropie ist die Ursache charakteristischer anamorphotischer Bildartefakte, insbesondere horizontal ausgeprägter Lens-Flares.

Für die Integration zylindrischer Linsen sind insbesondere zwei Aspekte relevant:

Intersection und Normalenberechnung

Die Schnittberechnung mit zylindrischen Linsenflächen unterscheidet sich grundlegend von der sphärischen Variante. Während sphärische Flächen durch eine rotationssymmetrische Kugelgleichung beschrieben werden, ergibt sich für zylindrische Flächen eine quadratische Gleichung, bei der die Krümmung ausschließlich in einer Raumrichtung wirksam ist.

Fokusberechnung und Astigmatismus

Ein zentrales Problem bei zylindrischen Linsen ist die Fokusberechnung. Durch den entstehenden Astigmatismus existiert kein singulärer Fokuspunkt, an dem beide Raumrichtungen gleichzeitig scharf abgebildet werden. Stattdessen ergeben sich unterschiedliche Fokuslagen für die horizontale und vertikale Achse.

Das in PBRT verwendete Fokusmodell basiert auf einer Thick-Lens-Approximation und setzt Rotationssymmetrie voraus. Diese Annahme ist für zylindrische oder anamorphotische Linsen nicht erfüllt und führt in der Praxis zu instabilen Fokuslagen sowie zu unscharfen oder verzerrten Lens-Flares.

Wahl des Referenzobjektivs

Für die konzeptionelle Auslegung der zylindrischen Linsenelemente wurde ein reales anamorphotisches Objektivdesign aus der Patentliteratur herangezogen. Als Referenz diente das europäische Patent **EP 3825750 A1**, das sich durch einen vergleichsweise übersichtlichen Aufbau auszeichnet.

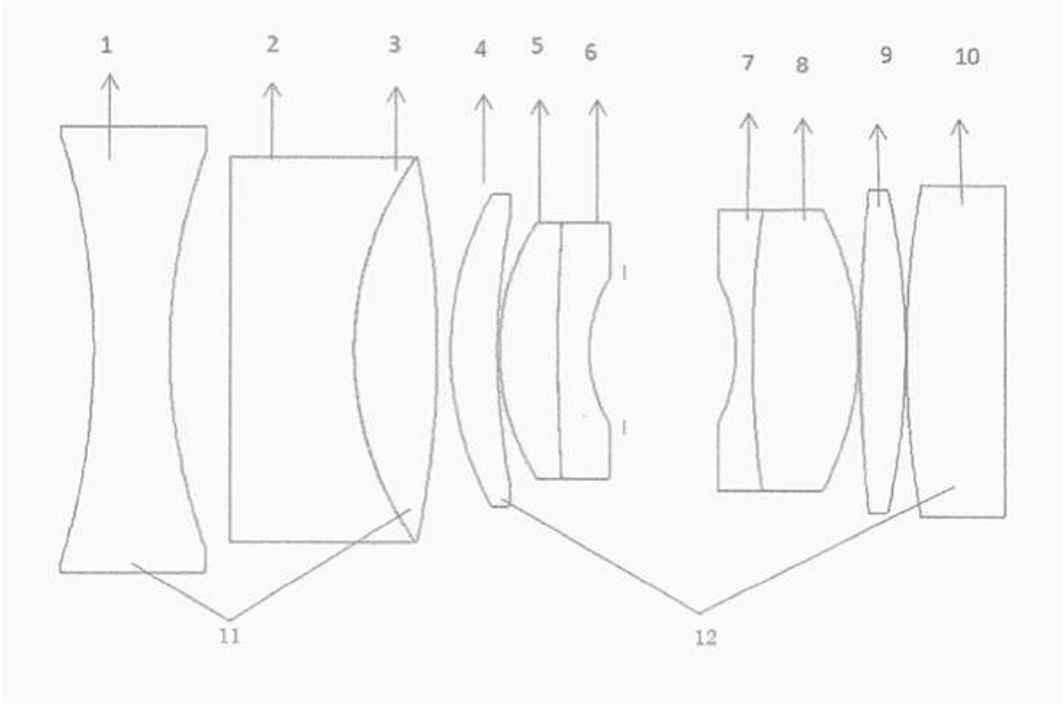


Abbildung X: Patent EP 3825750 A1

Lens	Surface Shape	radius (mm)	Thickness (mm)	Refractive index	Abbe Number	Mass (g)
First lens	Cylindrical	-68.330	7.460	1.5113	67.60	40.2
	Cylindrical	53.600	5.827			
Second lens	Cylindrical	inf	12.000	1.8325	20.11	60.4
Third lens	Cylindrical	31.100	8.000	1.9235	26.00	38.2
	Cylindrical	-88.300	1.260			
Fourth lens	Spherical	29.570	4.520	1.9108	35.25	16.5
	Spherical	66.000	0.220			
Fifth lens	Spherical	22.467	5.660	1.6968	55.53	8.5
Sixth lens	Spherical	203.000	3.030	1.6435	28.63	9.6
	Spherical	13.166	3.444			
Light bar			10.719			
Seventh lens	Spherical	-15.270	1.580	1.6612	27.24	4.8
Eighth lens	Spherical	95.000	10.350	1.8040	46.60	10.2
	Spherical	-27.650	0.200	-		
Ninth lens	Spherical	131.600	4.450	1.8040	46.60	8.2
	Spherical	-69.800	0.100			
Tenth lens	Spherical	89.800	9.420	1.8040	46.60	23.2
	Spherical	inf				

Abbildung Y: Optische Parameter (Tabelle)

Im Gegensatz zu anderen anamorphotischen Objektiven besteht dieses Design ausschließlich aus rein zylindrischen und sphärischen Linsen, ohne komplexe asphärische Flächen oder stark gekrümmte Speziallinsen. Die Anzahl der Linsenelemente ist überschaubar, und die geometrische Struktur ist klar definiert, was eine konzeptionelle Übertragung auf das implementierte Linsensystem erleichterte. Gleichzeitig

erzeugt die klare Gruppierung der zylindrischen Elemente die richtungsabhängige Abbildung, die für die anamorphotische Charakteristik entscheidend ist.

Praktische Umsetzung

In der Implementierung wird beim Ray–Lens-Intersection-Test abhängig vom Typ des jeweiligen `LensElementInterface` entschieden, welche Intersectionsroutine verwendet wird. Für sphärische Linsen kommt die klassische Kugelgleichung zum Einsatz, für zylindrische Linsen eine angepasste quadratische Gleichung mit anisotroper Krümmung.

Die Umsetzung in PBRT erwies sich als problematisch: Das rotationssymmetrische Fokusmodell von PBRT v4 unterstützt keine getrennte oder kombinierte Fokusoptimierung für die horizontalen und vertikalen Achsen. Bei anamorphotischen Linsen müsste für korrekten Fokus zudem der gesamte zylindrische Block verschoben werden, was in PBRT nicht implementiert ist. In der Praxis traten Laufzeitfehler auf, und nach Deaktivierung des Fokusmechanismus konnte die Linse nicht manuell scharf gestellt werden.

Daher konnten die zylindrisch beeinflussten Lens-Flares nicht korrekt simuliert werden. Trotz Übernahme des Patentdesigns blieb das Ergebnis praktisch unsichtbar, und die gewünschten Lens-Flare-Effekte traten im Renderbild nicht auf.

4.5 Kamera- und Szenenparameter für Lens-Flares

Zur Steuerung der Lens-Flare-Simulation wurden zwei zentrale Parameter in die Kamerakonfigurationn der Szene eingeführt.

1. **Anzahl der Flare-Samples**
    - Dieser Parameter legt fest, wie viele Strahlen insgesamt zur Berechnung der Lens-Flares erzeugt werden.
    - Die Samples werden gleichmäßig auf alle Lichtquellen in der Szene verteilt. Dabei erhält jede Lichtquelle eine bestimmte Anzahl lokaler Samples, wodurch eine gleichmäßige Abdeckung der Lichtquelle sowie der relevanten Linsenoberflächen gewährleistet wird.
    - Eine höhere Samplezahl reduziert das Monte-Carlo-Rauschen in den simulierten Flares, erhöht jedoch linear den Rechenaufwand.
2. **Chromatische Dispersion (Sellmeier vs. konstanter Brechungsindex)**
    - Dieser Boolean-Parameter steuert, ob die Berechnung der Brechungsindizes der Linsen wellenlängenabhängig über die Sellmeier-Gleichung erfolgt oder konstant bleibt.
    - Aktiviert man die Sellmeier-Option, wird für jede Wellenlänge des Strahls der spezifische Brechungsindex berechnet. Dies ermöglicht eine realistische Simulation chromatischer Farbverschiebungen in den Ghost-Artefakten.
    - Bei deaktivierter Dispersion wird ein fester Brechungsindex verwendet. Dadurch werden die Berechnungen beschleunigt, es entstehen jedoch keine Farbabweichungen in den Lens-Flares.

5 Ergebnisse

5.1 Messergebnisse

In den definierten Testszenen treten reproduzierbar Lens-Flares in Form von Ghost-Artefakten auf.

- **Positionsabhängigkeit:** Bei Lichtquellen nahe des Bildzentrums entstehen symmetrisch angeordnete Ghosts entlang der optischen Achse. Wird die Lichtquelle zum Rand verschoben, verlagern sich die Artefakte und werden durch das Objektivgehäuse beschnitten (Vignettierung).
    - *Zwei Lichtquellen (Abb. 5):* Überlappung der Flares.
    - *Lichtquelle befindet sich nicht im Bild (Abb. 3):* Nur Flares im Bild sichtbar
- **Apertur-Einfluss:** Die Form der Artefakte entspricht näherungsweise der Projektion der Blendenöffnung. Größere Aperturen führen zu weicheren Artefakten, während kleine Aperturen scharfe Konturen erzeugen.
- **Qualitätssteigerung:**
    - *Baseline (Abb. 6):* Überhöhte Intensität und starkes Rauschen.
    - *Fresnel-Korrektur (Abb. 7):* Realistische Intensitätsabnahme.
    - *Sampling (Abb. 8):* Konvergenz und Rauschunterdrückung durch erhöhte Samplezahlen.
    - *Spektrale Korrektur (Abb. 9):* Plausible Farbsäume (chromatische Aberration) ohne falsche Muster.

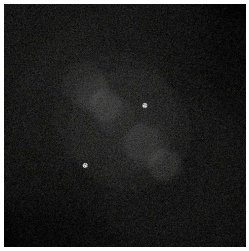


Abbildung 4: Two Lightsources

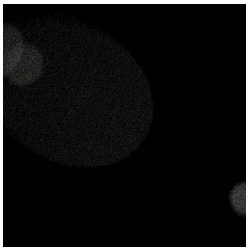


Abbildung 5: Lightsources out of the Picture

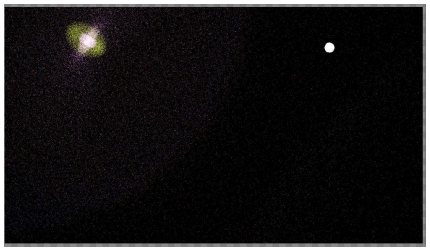


Abbildung 6: Baseline

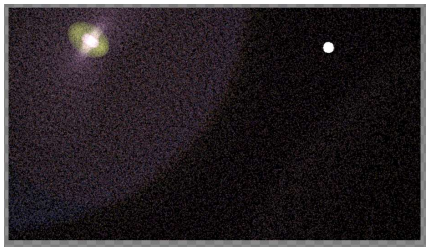


Abbildung 7: Mit Fresnel

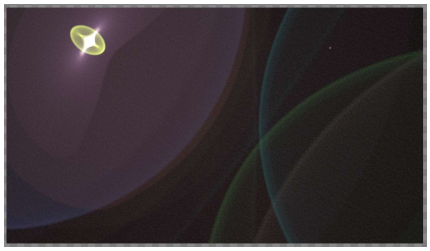


Abbildung 8: High Sampling

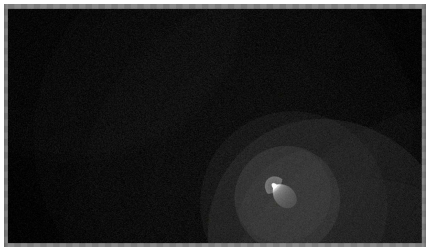


Abbildung 9: Spektral korrigiert

5.2 Auswertung

Die Qualität der gerenderten Lens-Flares skaliert direkt mit der Anzahl der Samples, entsprechend der typischen Monte-Carlo-Konvergenz. Die Laufzeit steigt nahezu linear mit der Anzahl der Flare-Samples.

Besonders auf der CPU ist das Rendering sehr zeitaufwändig, da für die physikalisch korrekte Simulation von Dispersion mehrere Wellenlängen-Samples pro Lichtquelle benötigt werden. Diese Mehrfachberechnung führt zu einer deutlichen Erhöhung der Renderzeit im Vergleich zu monowellenlängigen Approximationen mit denen wir anfangs gearbeitet haben.

6 Diskussion

6.1 Interpretation der Ergebnisse

Die implementierte Lösung zeigt, dass die Beschränkung auf Reflexionen zweiter Ordnung einen hervorragenden Kompromiss zwischen visueller Qualität und Rechenaufwand darstellt. Die Ghost-Artefakte decken sich phänomenologisch mit Beobachtungen an realen Objektiven. Die sichtbaren farbigen Ränder (Dispersion) bestätigen die korrekte Implementierung der Sellmeier-Gleichung. Diese Gleichung macht den Brechungsindex wellenlängenabhängig. Dadurch werden verschiedene Farben unterschiedlich stark gebrochen und reflektiert. Bei mehrfachen internen Reflexionen in der Linse (Lens-Flares) laufen Rot, Grün und Blau leicht unterschiedliche Wege, treffen an verschiedenen Positionen und mit unterschiedlicher Intensität auf den Sensor und das erzeugt leichte Farbsäume um die Flares. Das verbleibende Rauschen in den Flares ist inhärent für Monte-Carlo-Verfahren, bei denen nur ein kleiner Raumwinkel des Lichts tatsächlich den Sensor erreicht.

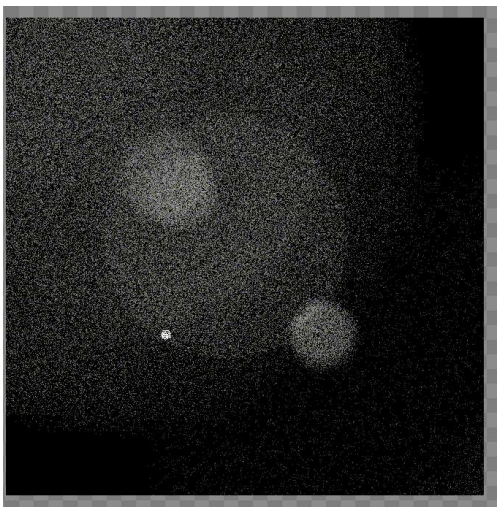


Abbildung 10: Ohne Sellmeier

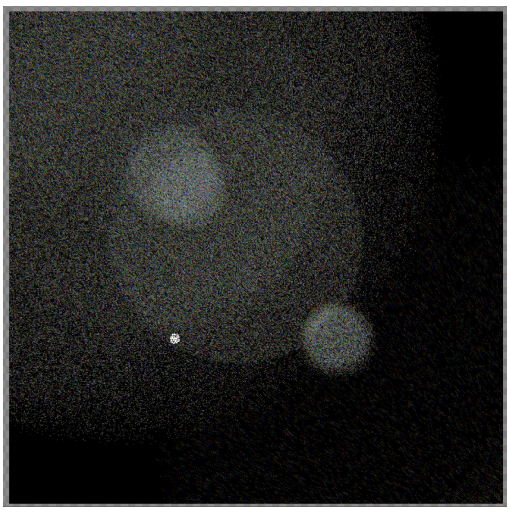


Abbildung 11: Mit Sellmeier

6.2 Fehlerbetrachtung

- **Sampling-Rauschen:** Trotz hoher Samplezahlen verbleibt hochfrequentes Rauschen in den Randbereichen der Flares. Hier könnten fortgeschrittene Sampling-Methoden (z.B. Metropolis Light Transport) Abhilfe schaffen.
- **Fehlende Beugung (Diffraktion):** Die aktuelle Implementierung basiert rein auf geometrischer Optik. Wellenoptische Effekte wie der "Starburst" an kleinen Blendenöffnungen werden nicht simuliert, was die Realitätsnähe bei sehr kleinen Blendenzahlen einschränkt.
- **Beschränkung der Reflexionstiefe:** Durch das Ignorieren von Reflexionen 4. Ordnung und höher könnten subtile Glüh-Effekte (Veiling Glare) fehlen, wenngleich deren energetischer Beitrag gering ist.

6.3 Nachteil bei reflektierenden Lichtquellen

Ein Nachteil des aktuellen Verfahrens besteht darin, dass die Lens-Flare-Simulation ausschließlich auf explizitem Light-Sampling basiert. Das bedeutet, dass nur solche Lichtquellen berücksichtigt werden, die im Szenengraph als Lichter definiert sind. Helle Bildbereiche, die durch indirekte Beleuchtung oder starke

Reflexionen entstehen, tragen hingegen nicht zur Erzeugung von Flares bei, obwohl sie aus physikalischer Sicht ebenfalls als effektive Lichtquellen wirken.

Das derzeitige Vorgehen entspricht damit funktional einer Form von *Light Tracing* bzw. *Photon Tracing* im Kameraraum: Strahlen werden gezielt von bekannten Lichtquellen durch das Linsensystem propagiert, wodurch Lens Flares ausschließlich dort entstehen, wo die Helligkeit der Szene direkt auf diese Lichtquellen zurückzuführen ist.

Ein vielversprechender theoretischer Ansatz zur Erweiterung dieses Modells ist eine hybride Sampling-Strategie, die zwei komplementäre Quellen für Flare-Samples kombiniert:

1. **Explizites Light-Sampling:** Wie bisher werden Strahlen von definierten Lichtquellen erzeugt und deterministisch durch das Linsensystem verfolgt. Dieser Pfad stellt sicher, dass klassische Lens-Flares, die durch starke Primärlichter verursacht werden, zuverlässig erfasst werden.

2. **Image-space guided Sampling:**
   Zusätzlich werden besonders helle Bildbereiche im bereits gerenderten Bild identifiziert, beispielsweise mithilfe eines Luminanz-Schwellenwerts oder einer bildbasierten Importance-Map. Diese Pixel werden als sekundäre, effektive Lichtquellen interpretiert, von denen aus wiederum Strahlen in Richtung Kamera bzw. durch das Linsensystem gesampelt werden.
   Auf diese Weise könnten auch Szenen berücksichtigt werden, in denen die visuell dominierende Helligkeit nicht von einer primären Lichtquelle ausgeht, sondern etwa von starken Spiegelungen auf Glasflächen, polierten Metalloberflächen, Wasserfläche* oder hochglänzenden Bodenmaterialien.

Die Kombination beider Strategien lässt sich konzeptionell als eine Form von Multiple Importance Sampling (MIS) verstehen:
Ein Teil der Flare-Strahlen wird aus der Verteilung der physikalisch definierten Lichtquellen gezogen, während ein weiterer Teil aus einer bildbasierten Importance-Verteilung stammt, die auf der tatsächlich im Bild auftretenden Helligkeit beruht. Dadurch könnten auch Lens-Flares durch indirekte Lichtquellen, stark gerichtete Sekundärreflexionen und spiegelnde Oberflächen erfasst werden, ohne die Effizienz des Verfahrens für klassische Lichtsituationen wesentlich zu beeinträchtigen.

7. Fazit und Ausblick

In dieser Arbeit wurde der Renderer PBRT v4 erfolgreich um eine physikalisch basierte Simulation von Lens Flares erweitert. Durch die Integration der Fresnel-Gleichungen und des Sellmeier-Dispersionsmodells in die `RealisticCamera` konnten realistische Ghost-Artefakte erzeugt werden. Die Ergebnisse validieren, dass die Berücksichtigung von Reflexionen 2. Ordnung für plausible Resultate ausreichend ist.

Die Erweiterung in Richtung zylindrischer beziehungsweise anamorphotischer Linsen erwies sich hingegen als deutlich anspruchsvoller. Obwohl die notwendige Geometrie und Datenstruktur konzeptionell umgesetzt werden konnten, zeigte sich, dass das in PBRT verwendete rotationssymmetrische Fokusmodell für anisotrope Linsensysteme nur eingeschränkt geeignet ist. Eine vollständige und stabile Abbildung anamorphotischer Effekte erfordert daher weiterführende Anpassungen der Fokusberechnung.

Ausblick: Zukünftige Arbeiten könnten sich auf die Simulation wellenoptischer Effekte konzentrieren, um Beugungsmuster an der Blende zu integrieren. Zudem wäre eine Portierung auf die GPU, beispielsweise unter Nutzung der PBRT-v4-GPU-Pipeline, sehr erstrebenswert, um interaktive Anwendungen zu ermöglichen. Für anamorphotische Linsen ist eine Weiterentwicklung der Fokus- und Autofokus-Strategie notwendig, um astigmatische Effekte robuster zu handhaben.

Eine weitere naheliegende Erweiterung ist die Berücksichtigung wellenlängenabhängiger Linsenvergütungen (Coatings). Reale Objektive verwenden Antireflexbeschichtungen, die den Fresnel-Reflexionsgrad abhängig von der Wellenlänge gezielt beeinflussen. Die Integration entsprechender Coating-Modelle würde die Simulation farbiger und materialspezifischer Lens Flares ermöglichen und die visuelle Übereinstimmung mit realen Kameras weiter verbessern.

Darüber hinaus könnten gezielte Importance-Sampling-Strategien für relevante Reflexionspfade sowie eine Validierung anhand realer Referenzaufnahmen die Effizienz und Aussagekraft der Methode weiter erhöhen.

8 Literaturverzeichnis

1. Hullin, M. B., Eisemann, E., Seidel, H.-P., & Lee, S. (2011). *Physically-Based Real-Time Lens Flare Rendering.* ACM Transactions on Graphics (SIGGRAPH).
2. Pharr, M., Jakob, W., & Humphreys, G. (2023). *Physically Based Rendering: From Theory to Implementation (v4).* MIT Press.
3. Hecht, E. (2016). *Optics* (5th ed.). Pearson.
4. Guangdong Sirui Optical Co., Ltd., *Anamorphic Lens*, European Patent EP 3 825 750 A1, veröffentlicht am 26. Mai 2021.
5. Teubner, U., & Brückner, H. J. (2019). *Optical Imaging and Photography: Introduction to Science and Technology of Optics, Sensors and Systems.* Berlin/Boston: De Gruyter. Kapitel 6.8. doi:10.1515/9783110472943
6. Blahnik, V., & Voelker, B. (2022). *About the reduction of reflections for camera lenses: How T*-coating made glass invisible.* Zeiss Lenspire. [Online]. Verfügbar unter: https://lenspire.zeiss.com/photo/app/uploads/2022/02/technical-article-about-the-reduction-of-reflections-for-camera-lenses.pdf
7. Pekkarinen, E., & Balzer, M. (2019). *Physically based lens flare rendering in The Lego Movie 2.* In *Proceedings of the 2019 Digital Production Symposium*, Los Angeles, CA, USA: ACM, pp. 1–3. doi:10.1145/3329715.3338881
8. Tang, H. (2011). *Rendering Realistic Lens Flare.* Technischer Projektbericht mit PBRT-basierter Implementierung.
9. Ansys, Inc. (o. J.). *Stray Light Analysis with Ghost Focus Generator.* [Online]. Verfügbar unter: https://optics.ansys.com/hc/en-us/articles/43071067483795-Stray-Light-Analysis-with-Ghost-Focus-Generator
10. Wikipedia. (o. J.). *Sellmeier-Gleichung.* [Online]. Verfügbar unter: https://de.wikipedia.org/wiki/Sellmeier-Gleichung

9 Anhang

Eigenständigkeitserklärung

Hiermit bestätigen wir, dass die hier vorliegende Arbeit selbstständig verfasst ist und nur die angegebenen Hilfsmittel genutzt wurden. Unter Angabe der Quellen wurde außerdem kenntlich gemacht, welche dem Wortlaut oder Sinn nach anderen Werken entnommen sind.

- **Marcel Kazemi**, 09.01.2026
- **Marvin Bernei**, 09.01.2026
- **Artem Grütter**, 09.01.2026