



MASTERTHESIS

zur Erlangung des akademischen Grades
Master of Engineering (M. Eng.)

BEWERTUNG DER EIGNUNG VERSCHIEDENER KAMERASYSTEME FÜR DEN HDR-WCG WORKFLOW

ANALYSE VON BROADCAST- UND CONSUMERKAMERAS
ANHAND AKTUELLER LUMINANZ- UND SPEKTRALMESSVERFAHREN

Hochschule	Hochschule der Medien Stuttgart
Fakultät	Electronic Media
Studiengang	Audiovisuelle Medien
Erstprüfer	Prof. Dr. Jan Fröhlich
Zweitprüfer	Prof. Stefan Grandinetti

Vorgelegt von	Jonas Christian Stolzmann
Matrikelnummer	42733
Datum	16.09.2024

Kurzfassung

Die Themenfelder High Dynamic Range (HDR) und Wide Color Gamut (WCG) haben in den 1990er Jahren in ersten Labortests Einzug in die Medienwelt erhalten. Spätestens seit 2015 sind die Themenfelder aus der Film- und Fernsehbranche nicht mehr wegzudenken. Viele Arbeiten haben sich in der Vergangenheit mit der Wahrnehmung und den Sichtbarkeitsgrenzen von HDR- und WCG-Produktionen auseinandergesetzt. In dieser Arbeit wird konkret auf die Bedingungen eingegangen, die Kamerasysteme erfüllen müssen, um für HDR-WCG-Produktionen geeignet zu sein. Dazu werden Messverfahren vorgestellt, um die Qualität verschiedener Kamerasysteme auf ihre Eignung für aktuelle und zukünftige HDR-WCG-Workflows sowie auf die Sichtbarkeitsgrenzen des visuellen Systems des Menschen zu überprüfen.

Insgesamt werden drei Broadcastkameras, drei Consumerkameras, eine Chipkamera und eine Drohne vermessen. In sieben empirischen Messreihen wird ein Qualitätsprofil erarbeitet, das die Luminanz- und Spektraleigenschaften der Kamerasysteme zeigt. Zu den Eigenschaften zählen die Gammakurve, die Full-Well-Capacity, die Vignettierung, das Rauschen, der Signal-Noise-Ratio (SNR), die Dark Signal Non-Uniformity (DSNU), die Photo-Response Non-Uniformity (PRNU) und die Farbtreue.

Mithilfe der Messergebnisse wird gezeigt, unter welchen Voraussetzungen die Kamerasysteme die Grenzen der Wahrnehmung des visuellen Systems erreichen. Darüber hinaus wird gezeigt, welche Bedingungen Kamerasysteme erfüllen müssen, um für die aktuellen HDR-WCG-Workflows Perceptual Quantizer (PQ) und Hybrid Log Gamma (HLG) geeignet zu sein.

Abstract

The fields of High Dynamic Range (HDR) and Wide Color Gamut (WCG) first appeared in the media world during laboratory tests in the 1990s. Since 2015, these topics have become indispensable in the film and television industry. Numerous studies in the past have addressed the perception and visibility limits of HDR and WCG productions. This work specifically focuses on the conditions that camera systems must meet to be suitable for HDR-WCG productions. Measurement methods are presented to evaluate the quality of various camera systems in terms of their suitability for current and future HDR-WCG workflows, as well as their adherence to the visibility limits of the human visual system.

A total of three broadcast cameras, three consumer cameras, a chip camera, and a drone are measured. In seven empirical measurement series, a quality profile is developed, which highlights the luminance and spectral characteristics of the camera systems. These characteristics include the gamma curve, full-well capacity, vignetting, noise, Signal-Noise-Ratio (SNR), Dark Signal Non-Uniformity (DSNU), Photo-Response Non-Uniformity (PRNU), and color accuracy.

Using the measurement results, the conditions under which the camera systems meet the perceptual limits of the human visual system are demonstrated. Additionally, the conditions that camera systems must meet to be suitable for the current HDR-WCG workflows Perceptual Quantizer (PQ) and Hybrid Log Gamma (HLG) are outlined.

Danksagung

Es heißt, dass es ein ganzes Dorf braucht, um ein Kind großzuziehen. Ob das stimmt, weiß ich nicht, aber ich finde es im Falle meiner Masterthesis sehr zutreffend. Ohne die Unterstützung vieler verschiedener Menschen wäre diese Arbeit nicht zu dem gewachsen, was sie ist. Daher möchte ich mich ganz herzlich bei all denen bedanken, die mich bei meiner Masterarbeit und auf dem Weg dahin unterstützt haben. Vielen Dank an ...

- Dr. Jan Fröhlich und Stefan Grandinetti, für eure Expertise und Geduld, die wertvollen Anregungen und hier und da kritische Fragen über den gesamten Forschungsprozess hinweg.
- Justus Mai und India Fleuchaus für eure Hilfe und Unterstützung bei der Auswertung und für die vielen Telefonate bei Detailfragen.
- Ayla Remmele, Daniel Kuner, Paul Winterstein und Manuel Josia Hildebrandt für die vielen Co-Working-Sessions.
- Lea Piaskowski, Priska Merbitz-Zahradnik für eure Unterstützung im Schreibprozess und für eure Zeit und Unterstützung bei der Korrektur.

Mein besonderer Dank gilt auch den Menschen und Unternehmen, die den Forschungsaufbau möglich gemacht haben und mir neben teurem Equipment und empfindlichen Messinstrumenten immer mit Rat und Tat zur Seite gestanden haben. Vielen Dank an ...

- Frank Zellner und Peter Ruhrmann, dass ihr mich von Seiten der Hochschule mit allem nötigen Equipment ausgestattet habt.
- Sven Bauschke und Kamilla Wysocki sowie die Lang AG, für eure inhaltliche und materielle Unterstützung und die Möglichkeit meinen Forschungsaufbau in Lindlar bei euch durchführen zu können.
- Claudia Rosenkranz und Dietmar Wüller sowie die Firma Image Engineering für die Unterstützung durch das Testchart.

- Jörg Lehnert und Artur Bayer sowie die Firma Qvest für das zur Verfügung stellen der Ulbrichtkugel.
- Max Bremen und Phillip Nöcker-Prior für eure Unterstützung bei der Organisation des Equipments in den Firmen und Markus Neeb für das Stellen der Consumerkameras sowie euch drei für all eure Unterstützung auch abseits der Masterthesis.

Zum Schluss möchte ich mich von Herzen bei all denen bedanken, die mich über den Prozess der letzten Jahre begleitet und die Jahre meines Studiums zu einer unvergesslichen Zeit gemacht haben. Ganz besonders gilt mein Dank meinen Eltern und Geschwistern.

Mama, Papa, Tobias und Anna: Danke für euren unerschütterlichen Glauben an mich und eure bedingungslose Unterstützung in allen Phasen meines Studiums und weit darüber hinaus. Ohne euch hätte ich diese Herausforderung nicht gemeistert. Von ganzem Herzen:

Danke

Inhaltsverzeichnis

1. Einleitung	1
1.1. Problemstellung und Relevanz des Themas	1
1.2. Forschungsfragen	2
1.3. Aufbau und Struktur der Masterarbeit	3
2. Grundlagen des Human Visual System	5
2.1. Optische Eigenschaften des Lichts	5
2.1.1. Eigenschaften der Wellen- und Strahlenoptik	6
2.1.2. Grundgrößen der Fotometrie	9
2.1.3. Spektrale Eigenschaften des Lichts	10
2.2. Der Aufbau des menschlichen Auges	11
2.3. Wahrnehmung des HVS	13
2.3.1. Helligkeits- und Kontrastwahrnehmung	13
2.3.2. Nichtlinearität des HVS	14
2.3.3. Rauschwahrnehmung	15
2.3.4. Farbwahrnehmung	16
2.4. Sichtbarkeitsgrenzen des HVS	17
3. High Dynamic Range und Wide Color Gamut	19
3.1. High Dynamic Range	19
3.1.1. Quantisierung	20
3.1.2. Szenenreferenziert vs. Displayreferenziert	20
3.1.3. Transferfunktionen	21
3.2. Wide Color Gamut	23
3.2.1. CIE-1931-Normfarbsystem	23
3.2.2. Farbräume	24
3.2.3. Farbraumtransformation	26
3.2.4. Weißpunkttransformation	27
3.2.5. Vergleich von Farbwerten	28
3.3. Vorgaben an Kamerasysteme	29
4. Kameratechnik	33
4.1. Eigenschaften elektrischer Kameraelemente	33
4.1.1. Bildwandler	33
4.1.2. Abbildungsfehler elektrischer Kameraelemente	36
4.2. Eigenschaften optischer Kameraelemente	39
4.2.1. Optik	39
4.2.2. Abbildungsfehler optischer Kameraelemente	41
4.3. Qualitätsmerkmale von Kamerasystemen	42

5. Messung	45
5.1. Methodik	45
5.2. Messaufbau	46
5.2.1. Messgeräte	48
5.2.2. Kamerasysteme	49
5.3. Messansatz	50
5.4. Messdurchführung	51
5.4.1. Verifikation der Messinstrumente	51
5.4.2. Luminanzmessungen der Kameras	53
5.4.3. Spektralmessung der Kameras	56
6. Auswertung	57
6.1. Verifikation der Messgeräte	57
6.1.1. Auswertung 1: Verifikation der Linearität der Lichtquelle . . .	57
6.1.2. Auswertung 2: Verifikation der Farbkonstanz der Lichtquelle	60
6.1.3. Auswertung 3: Verifikation des Messcharts TE223	61
6.1.4. Auswertung 4: Verifikation des Messcharts TE188	62
6.2. Kameraanalyse	65
6.2.1. Auswertung 5: Gammakurve und Full-Well-Capacity	65
6.2.2. Auswertung 6: Vignette	69
6.2.3. Auswertung 7: Rauschen	71
6.2.4. Auswertung 8: Signal-Rauschabstand und Dynamikumfang .	73
6.2.5. Auswertung 9: DSNU	76
6.2.6. Auswertung 10: PRNU	78
6.2.7. Auswertung 11: Farbtreue	79
6.3. Interpretation der Daten	82
6.3.1. Anforderung 1: Wahrnehmungsgrenze des Steady State . . .	83
6.3.2. Anforderung 2: Wahrnehmungsgrenze des Rauschens	83
6.3.3. Anforderung 3: Farbdifferenzen zum Referenzfarbraum . . .	84
6.3.4. Anforderung 4: Farbdifferenzen der Kamerasysteme	85
6.3.5. Anforderung 5: SDR-Workflow	85
6.3.6. Anforderung 6: HLG-Workflow	86
6.3.7. Anforderung 7-9: PQ-Workflow bei 1000, 4000 und 10000 cd/m ² Spitzenleuchtdichte	86
7. Abschluss	89
7.1. Fazit	89
7.2. Limitationen	90
7.3. Ausblick	91
A. Anhang A: Herstellerangaben	XVII
A.1. Herstellerangaben des TE223	XVII
A.2. Herstellerangaben des TE188	XVIII
B. Anhang B: Messungen	XIX
B.1. Messung 1	XIX
B.2. Messung 2	XXV

B.3. Messung 3	XXXI
B.4. Messung 4	XXXII
B.5. Messung 5	XXXIII
B.6. Messung 6	XXXIV
B.7. Messung 7	XXXIX
C. Anhang C: Auswertungen	XLI
C.1. Auswertung 5	XLI
C.2. Auswertung 6	XLIII
C.3. Auswertung 7	XLVI
C.4. Auswertung 8	XLVIII
C.5. Auswertung 9	LI
C.6. Auswertung 10	LIII
D. Anhang D: Quellcode	LVII
D.1. Quellcode Auswertung 1	LVII
D.2. Quellcode Auswertung 2	LXI
D.3. Quellcode Auswertung 3	LXV
D.4. Quellcode Auswertung 4	LXVII
D.5. Quellcode Auswertung 5	LXXI
D.6. Quellcode Auswertung 6	LXXIX
D.7. Quellcode Auswertung 7	LXXXIII
D.8. Quellcode Auswertung 8	LXXXVIII
D.9. Quellcode Auswertung 9, 10	XCII
D.10. Quellcode Auswertung 11	C

1. Einleitung

1991 entwickelte Georges Cornu jols ein Verfahren, das als Meilenstein in der digitalen Bildverarbeitung angesehen werden kann. Das Patent „Device for increasing the dynamic range of a camera“ beschreibt eine Methode zur Aufnahme, Verarbeitung und Wiedergabe von Bildern mit erweitertem Dynamikbereich. Durch die Kombination von Bilddaten aus zwei Bildsensoren wurde aus zwei Einzelbildern mit unterschiedlicher Belichtung eine Bildkomposition mit erweitertem Dynamikumfang erstellt. Diese Entwicklung ermoglicht es, sehr helle und sehr dunkle Bereiche eines Bildes darzustellen, was in herkommlichen Aufnahme- und Wiedergabeverfahren bis dahin nicht moglich war. [9]

1.1. Problemstellung und Relevanz des Themas

Durch den Fortschritt in der Kameratechnik sind heutige Kamerasensoren leistungsfahiger, farbtreuer und weniger rauschempfindlich als 1991. Dennoch konnen sie nicht den vollen Dynamikumfang und Farbbereich abbilden, der mit dem menschlichen Auge wahrgenommen werden kann. In den Themenfeldern High Dynamic Range (HDR) und Wide Color Gamut (WCG) wird versucht, diese Barriere in der Bildverarbeitung zu minimieren, mit dem Ziel eine Szene von der Aufnahme bis zur Wiedergabe exakt so darzustellen, wie sie mit dem Auge wahrgenommen werden kann [13].

Aufnahmeseitig unterscheiden sich Kameratypen haufig in den verbauten Sensoren und Ausleseelektroniken. Zudem werden Sensordaten unterschiedlich vorverarbeitet und charakterisiert. Mit dem Einsatz von Debayering- und Kompressionsalgorithmen, Rauschunterdruckung, Farbmatrizen und Lumakurven unterscheidet sich auch die Interpretation dieser Sensordaten. Dadurch konnen Kameras in ihrer Farbdarstellung, dem Rauschverhalten und ihrem Dynamikumfang stark von der Wahrnehmung des Menschen und von anderen Kamerasystemen abweichen. Auch die Szene hat Einfluss auf die Qualitat der Abbildung. Weicht die Helligkeit einer Szene zu stark von dem nativen ISO eines Kamerasystems ab, verschlechtert sich folglich der Signal-Rauschabstand. Damit andert sich der Dynamikumfang mit den Anforderungen an eine Szene [51, S. 398-399].

Wiedergabeseitig mussen ebenfalls Kompromisse gemacht werden. Auch wenn gemaf des Branchenkompass fur das erste Quartal 2024 bereits 100 Prozent aller neu verkauften TV-Gerate HDR fahig sind, nutzen viele Konsumenten weiter Ge-

räte, die Standard Dynamic Range (SDR) und Standard Color Gamut (SCG) Content erwarten. Ein SDR- und SCG-Workflow bleibt also mittelfristig unabdingbar. [12]

Obwohl Fernseher in den letzten Jahren immer leistungsfähiger wurden, werden Spitzenleuchtdichten, die die Wahrnehmungsgrenze des Human Visual System (HVS) erreichen, aktuell nur unter Laborbedingungen erzielt. Gängige Konsumergeräte erreichen aktuell eine Spitzenhelligkeit von bis zu 2000 cd/m² und können rund 45,5 Prozent der für das menschliche Auge sichtbaren Farben abbilden. [51, S. 30-32] [52]

Mit der Vielzahl an Herstellern von Kameras und TV-Geräte steigt auch die Herausforderung bei der Integration in einen gemeinsamen Workflow. Abbildung 1.1 zeigt eine klassische Produktionspipeline für Fernsehproduktionen.

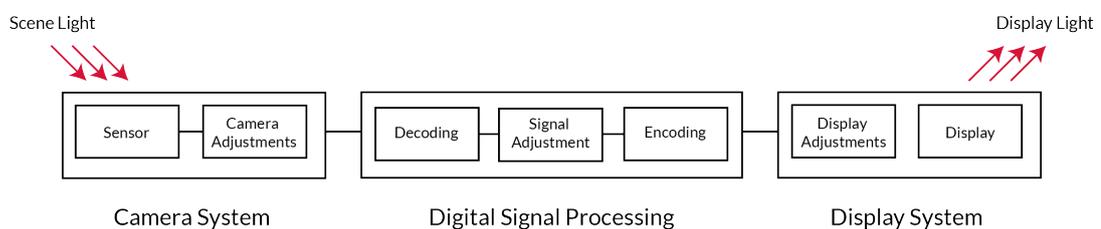


Abbildung 1.1.: Bildgebungspipeline, frei nach [22, Abb. 1.1], [13, Abb 2.1], [57, Abb. 3] und [23, Abb. 1]

Eine TV-Produktion gelingt nur dann, wenn alle Kamera- und Wiedergabesysteme die Produktionsanforderungen erfüllen und von den Bildverarbeitungs- und Transfersystemen richtig interpretiert und weiterverarbeitet werden können. Sie unterscheiden sich jedoch je nach Fernsehformat erheblich. Während bei Studio-produktionen und szenischen Formaten die Beleuchtungssituation gut kontrolliert werden kann, können in Liveproduktionen und dokumentarischen Produktionsformaten die Produktionsbedingungen mitunter stark abweichen.

1.2. Forschungsfragen

In der vorliegenden Forschungsarbeit werden Anforderungen an aktuelle Kamerasysteme definiert, die für einen HDR-WCG Workflow in Live-Produktionen gelten und verschiedene Kamerasysteme auf diese Anforderungen überprüft. Die Anforderungen werden auf Grundlage verschiedener Wahrnehmungsaspekte des HVS und aktueller HDR - WCG Standards festgelegt.

Daraus ergeben sich die folgenden Forschungsfragen für diese Arbeit:

- Welche Anforderungen lassen sich aufgrund der visuellen Wahrnehmung des Menschen an Kamerasysteme definieren?

- Welche spezifischen Anforderungen stellen aktuelle HDR-WCG Standards an Kamerasysteme?
- Welche Kameras aus dem aktuellen Produktionsumfeld erfüllen die Anforderungen an die visuelle Wahrnehmung des Menschen und die Anforderungen für HDR-WCG-Produktionen?

1.3. Aufbau und Struktur der Masterarbeit

Um die Fortschritte in der digitalen Bildverarbeitung einzuordnen, wird in der vorliegenden Masterarbeit zunächst auf das HVS eingegangen. Dabei werden die physiologischen und psychologischen Aspekte der menschlichen Farb-, Helligkeits- und Rauschwahrnehmung erläutert 2.

Anschließend werden die technischen Grundlagen derameratechnik und die spezifischen Merkmale von HDR-WCG-Produktionen erläutert und Anforderungen an dieameratechnik für den Live-Workflow abgeleitet 3 4.

Im experimentellen Teil dieser Arbeit wird ein Testaufbau entwickelt mit dem verschiedene Kameras auf Grundlage dieser Anforderungen überprüft werden. Dabei wird dargestellt, wie gut moderne Kameras in der Lage sind, die erweiterten Möglichkeiten von HDR und WCG Produktionen auszuschöpfen 5.

Abschließend werden die Ergebnisse des Testaufbaus diskutiert und ausgewertet. Schlussfolgerungen werden dahingehend gezogen, welche Bedingungen erfüllt sein müssen, damit bestimmte Kamerasysteme für einen HDR-WCG Workflow geeignet sind. Zudem werden die Limitationen der Arbeit erläutert 6.

2. Grundlagen des Human Visual System

Die visuelle Wahrnehmung des Menschen entsteht aus dem Zusammenspiel der physikalischen Eigenschaften des Lichts, den physiologischen Eigenschaften des menschlichen Auges und der Verarbeitung visueller Informationen im Gehirn. In diesem Grundlagenkapitel werden die wesentlichen Eigenschaften des menschlichen visuellen Systems betrachtet. Dazu wird zunächst auf die optischen Eigenschaften des Lichts eingegangen. Anschließend wird der Aufbau des menschlichen Auges und der Einfluss auf das Sehen erläutert. Abschließend kommt ein umfangreicher Abschnitt zur Wahrnehmung in Bezug auf die Helligkeit, die Farben und das Rauschen einer Szene und welche Modelle sich auf dessen Grundlage ergeben. Abbildung 2.1 zeigt wie das Kapitel strukturiert ist.

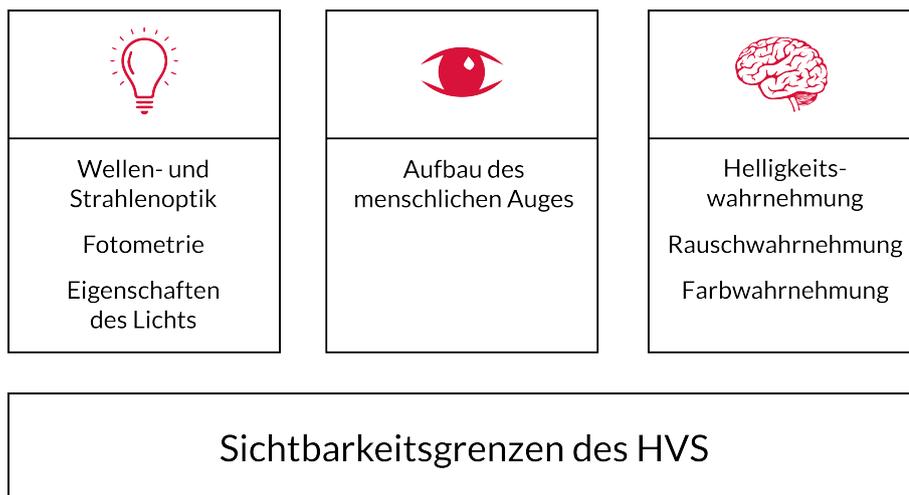


Abbildung 2.1.: Übersicht zum Grundlagenkapitel Human Visual System

2.1. Optische Eigenschaften des Lichts

Bereits 1678 zeigte Christiaan Huygens, dass sich einige Phänomene der Ausbreitung von Licht erklären lassen, wenn man davon ausgeht, dass sich Licht im Raum in einer Wellenform ausbreitete. In dem Huygensschen Prinzip hat er formuliert, dass jeder Punkt einer Welle als Ausgangspunkt einer neuen Welle betrachtet werden kann. Die Überlagerung begründet die Ausbreitungsrichtung der Lichtwellen. Das Huygenssche Prinzip gilt als Ausgangspunkt für das Feld der Wellenoptik. [32]

1905 hat Albert Einstein die Lichtquantenhypothese aufgestellt. Demnach wird

Licht als quantisiertes Lichtteilchen (Photon) beschrieben. Ein Lichtstrahl ist nach dieser Definition ein Photonenstrom bei dem jedes Photon ein Energiequant ist. Die Energie des Photons ist dabei abhängig von der Frequenz des Lichts. Sie ist in folgender Notation definiert [56, S.1007 - 1032] [28]:

$$E = \frac{hc}{\lambda} = h * \nu \quad (2.1)$$

Damit hat Einstein die Vorstellung revolutioniert, dass Licht eine elektromagnetische Welle sei und nur als solche interpretiert werden kann.

1927 haben Niels Bohr und Werner Heisenberg die Kopenhagener Deutung formuliert. In dem darin enthaltenen Komplementaritätsprinzip von Niels Bohr wurde festgelegt, dass zwei verschiedene Beobachtungen eines Vorgangs sich gegenseitig ausschließen und dennoch ergänzen können. Das zunächst widersprüchlich wirkende Prinzip entstand aus dem Kompromiss, dass sich einige Eigenschaften in der Quantenphysik nur über die Eigenschaften klassischer Wellen erklären lassen, andere jedoch nur über die Eigenschaften klassischer Teilchen. Ob Licht eine Welle oder ein Teilchen ist lässt sich bis heute nicht beantworten. Daher spricht man von dem Wellen-Teilchen-Dualismus [56, S.1007 - 1032] [47].

Einige der für diese Arbeit relevanten Eigenschaften des Lichts werden im Folgenden erläutert und lassen sich teilweise über die Eigenschaften klassischer Teilchen (Strahlenoptik) und teilweise über die Eigenschaften klassischer Wellen (Wellenoptik) erklären.

2.1.1. Eigenschaften der Wellen- und Strahlenoptik

Trifft ein Lichtstrahl auf eine Fläche, so wird ein Teil der Photonen reflektiert. Bei einer gerichteten Rückstrahlung des Lichts spricht man von **Reflexion**. Nach dem Reflexionsgesetz entspricht der Einfallswinkel dem Ausgangswinkel des Photonenstrahls in Relation zum Normalenvektor der Oberfläche [56, S.1011 - 1018] [8, S. 49].

Bei einer diffusen Rückstrahlung des Lichts spricht man von **Remission**. Ursache für die Remission ist eine ungeordnete Oberflächenstruktur an der die Photonen in alle Richtungen in den Raum reflektiert werden [51, S. 18]. Bei der Rückstrahlung von Licht kommt es immer zu Remission und Reflexion. Je glatter die Oberflächenstruktur desto höher ist der reflektierte Anteil [8, S. 49].

Trifft der Lichtstrahl auf eine Grenzfläche zwischen zwei unterschiedlichen Medien, dann werden Anteile des Lichts reflektiert und der andere Teil tritt in das Medium ein. Innerhalb unterschiedlicher Medien hat Licht eine unterschiedlich starke

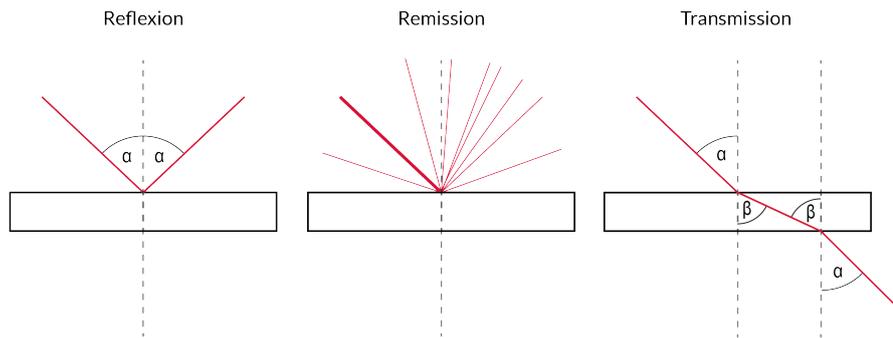


Abbildung 2.2.: Reflexion, Remission und Brechung von Licht, frei nach [51, Abb. 2.4 und 2.6].

Ausbreitungsgeschwindigkeit. Dadurch kommt es am Übergang zur **Brechung** des Lichts. Je dichter ein optisches Medium ist, desto langsamer breitet sich das Licht in diesem aus. Abbildung 2.2 zeigt die Reflexion, die Remission und die Brechung an einem optisch dünneren Medium. [56, S.1011 - 1014]



Abbildung 2.3.: Albumcover des Albums „Dark Side of the Moon“ der Band Pink Floyd aus dem Jahre 1973 [2]

Die Stärke der Brechung durch ein Medium ist abhängig von der Wellenlänge des einfallenden Lichts. Dessen Abhängigkeit wird als Dispersion bezeichnet und kann sehr anschaulich auf dem Album Cover von „Dark Side of the Moon“ der Band Pink Floyd aus dem Jahre 1973 betrachtet werden (siehe Abbildung 2.3). Der Brechungswinkel von einfallendem Licht ist umso kleiner, je kurzwelliger das Licht ist. Die Dispersion ist ein wichtiger Effekt und findet vor allem Anwendung in der Spektralanalyse [56, S.1019 - 1021].

Bei der Interaktion von Licht mit Atomen kommt es zur **Absorption**. Bei der Absorption nimmt ein Elektron die Energie des Photons auf, wodurch das Elektron angeregt wird. Das Elektron springt dabei auf ein höheres Energieniveau in der Atomhülle. Die Atome gehen in einen angeregten Zustand über. Ein Elektron wird nur dann angeregt, wenn die Energie des Photons der Energiedifferenz zwischen dem Grundzustand des Elektrons und dem angeregten Energieniveau entspricht. Da die Energie von der Frequenz des Photons abhängig ist, werden in Abhängigkeit zur Oberfläche nur bestimmte Frequenzen des Lichts absorbiert [28, S. 524].

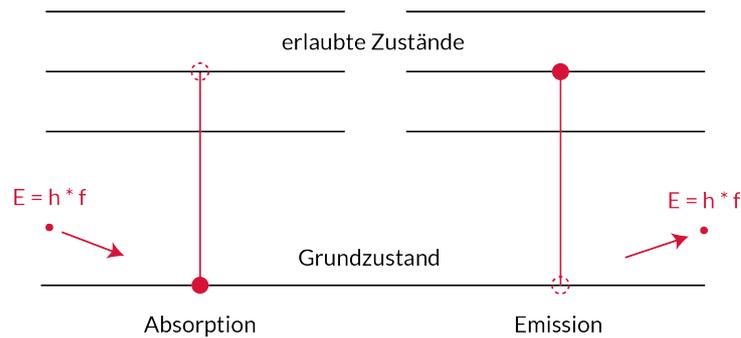


Abbildung 2.4.: Absorption und Emission von Photonen, frei nach [51, Abb. 2.3]

Die Rückreaktion der Absorption wird als **Emission** bezeichnet. Dabei springt das Elektron aus dem angeregten Zustand in den Grundzustand zurück und emittiert dabei ein Photon in beliebiger Raumrichtung [28, S. 524].

Zwischen Lichtphotonen und Atomen kann es zu Wechselwirkungen kommen, wobei die Photonen von ihrer geradlinigen Bahn abgelenkt werden. Diese Abweichung nennt man **Streuung**. Die Intensität der Ablenkung des Lichts steht in Abhängigkeit zur Wellenlänge des Lichts. Aus dieser Abhängigkeit zur Wellenlänge lassen sich viele Naturphänomene erklären. Sie ist auch der Grund dafür, warum der Himmel am Tag blau erscheint. Mit wechselnden Wetterbedingungen werden unterschiedliche Wellenlängen unterschiedlich stark gestreut, was zu stetig wechselnden Lichtverhältnissen führt [28, S. 481-487] [8, S. 50].

Der **Photoelektrische Effekt** beschreibt Wechselwirkungen zwischen den Elektronen eines Halbleitermaterials und den Photonen des Lichts. Es lässt sich zwischen dem äußeren und dem inneren Photoelektrischen Effekt unterscheiden. Der äußere photoelektrische Effekt tritt dann auf, wenn ausreichend Photonen mit passender Wellenlänge auf ein Metall oder einen Halbleiter wirken. Durch die Absorption der Photonen wird dabei so viel Energie übertragen, dass Elektronen aus der Atomhülle ausgestoßen werden. Als Reaktionsprodukt entsteht ein ionisiertes Atom und ein freies Elektron. Bei dem inneren photoelektrischen Effekt werden Photonen absorbiert, bei denen Elektronen innerhalb der Atomhüllen in einen erhöhten Zustand übergehen ohne das Atom zu verlassen. Durch diese Anregung bilden sich Elektronen-Loch-Paare, die die elektrische Leitfähigkeit des Materials beeinflussen. Dieser innere und äußere photoelektrische Effekt wird in Fotozellen genutzt, um die Photonenmenge, die auf einen Sensel trifft, zu messen [31, S.1 ff.].

Bei der Ablenkung von Lichtwellen an einem Hindernis, kommt es zur **Beugung (Diffraktion)**. Sie entsteht immer dann, wenn die Ausbreitungsrichtung einer Lichtwelle versperrt ist. Gemäß des Huygensschen Prinzips kann jeder Punkt als Ausgangspunkt einer neuen Welle betrachtet werden, sodass es an den Kanten des Hindernisses zur kugelförmigen Ausbreitung der neuen Welle kommt. Dadurch breitet sie sich in alle Richtungen aus. Hinter dem Hindernis kommt es zur Überlagerung von Wellen, die eine neue Wellenfront bilden und so die Ausbreitungsrichtung des Lichts beeinflussen. Bei Kameraobjektiven beeinflusst die Beugung

des Lichts das Auflösungsvermögen, wodurch die mögliche Blendenöffnung des Objektivs begrenzt ist [8, S. 48-49].

Einzelne Wellen überlagern sich durch die Addition ihrer Amplituden, was die Intensität der Amplitudenwerte beeinflusst. Diese Überlagerung wird als **Interferenz** bezeichnet. Verlaufen zwei sich überlagernde Wellen genau gegenläufig, so kommt es zur destruktiven Interferenz, also der Auslöschung beider Wellen. Verlaufen sie gleichläufig, so verstärken sich die Wellen in ihren Minima und Maxima. Das wird als konstruktive Interferenz bezeichnet [8, S. 48].

An dieser Stelle gilt es zu erwähnen, dass es eine Vielzahl weiterer Eigenschaften von Licht in der Wellen- und Strahlenoptik gibt, wie beispielsweise die Polarisation. Diese spielen für die relevanten Effekte in dieser Arbeit jedoch eine untergeordnete Rolle und werden deshalb nicht detaillierter betrachtet.

2.1.2. Grundgrößen der Fotometrie

Bei der Fotometrie geht es um die Messung von Lichtintensitäten und Helligkeiten. In diesem Abschnitt soll es um die energetischen Grundgrößen gehen, die für die Messung von Luminanzen notwendig sind.

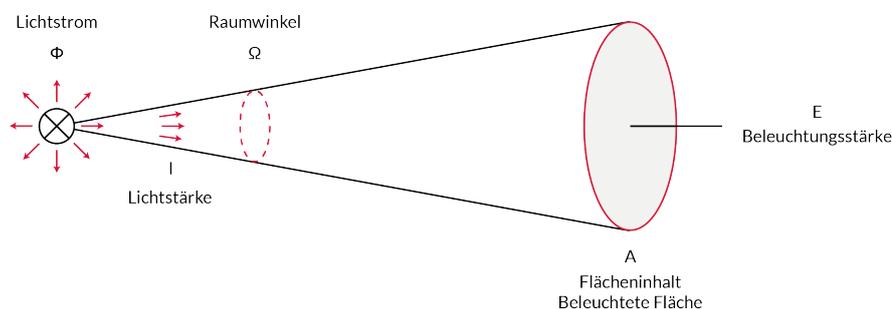


Abbildung 2.5.: Grundgrößen der Fotometrie

Abbildung 2.5 zeigt die fotometrischen Grundgrößen. Der **Lichtstrom** ϕ beschreibt die Lichtenergie, die von einer Lichtquelle pro Zeit ausgesandt wird. Er wird in Lumen (lm) gemessen und ist ein Maß für die Helligkeit einer Lichtquelle. Geht man von einer gerichteten Abstrahlung aus, so spricht man von der **Lichtstärke** I . Die Lichtstärke kann aus dem gemessenen Lichtstrom und dem Raumwinkel Ω berechnet werden. Die Formel für die Berechnung lautet [51, S.20-21]:

$$I = \frac{\phi}{\Omega} \quad (2.2)$$

Die gemessene Helligkeit einer beleuchteten Fläche wird als **Beleuchtungsstärke** E bezeichnet. Sie ist definiert als das Verhältnis aus Lichtstrom zur beleuchteten

Fläche und lässt sich wie folgt berechnen [51, S.20-21]:

$$E = \frac{I}{A * \cos \epsilon} = \frac{R * E}{\pi} \quad (2.3)$$

Mit dem steigenden Abstand D zwischen der Lichtquelle und der beleuchteten Fläche A nimmt die Beleuchtungsstärke mit dem Quadrat der Entfernung ab. Verdoppelt sich also der Abstand, wird für die gleiche Beleuchtungsstärke die vierfache Lichtstärke benötigt. Abbildung 2.6 zeigt den Zusammenhang der Entfernung zur Beleuchtungsstärke. Er wird als das **photometrische Entfernungsgesetz** bezeichnet und hat einen erheblichen Einfluss auf die Messung von Beleuchtungsstärken, da man vordefinierte Abstände genau einhalten muss, um belastbare Ergebnisse zu erhalten.

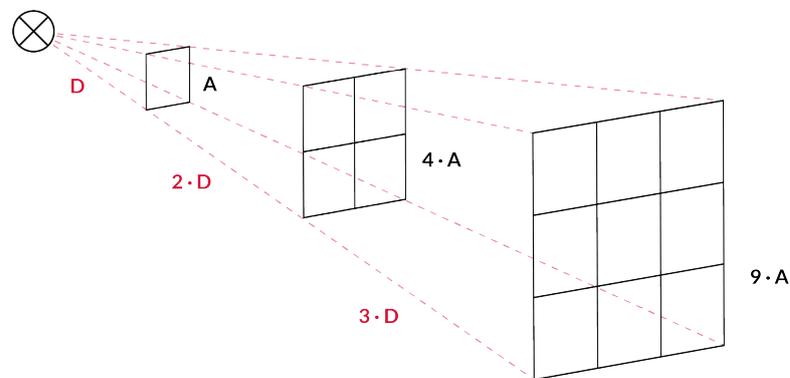


Abbildung 2.6.: Visualisierung des photometrisches Entfernungsgesetzes

Die **Leuchtdichte** L beschreibt die Helligkeit einer leuchtenden oder beleuchteten Fläche in Abhängigkeit von ihrer Richtung. Sie ist definiert als Lichtstärke pro Flächeneinheit und wird in cd/m² angegeben. Im Englischen wird sie häufig auch in Nits angegeben. Die Leuchtdichte berechnet sich wie folgt [51, S.20-21]:

$$L = \frac{\phi}{A} = \frac{I * \cos \epsilon * \Omega}{r^2} \quad (2.4)$$

2.1.3. Spektrale Eigenschaften des Lichts

Die Verteilungen von Lichtintensitäten über verschiedene Wellenlängen hinweg beschreiben die **spektralen Eigenschaften** von Licht. Sie können durch die Wahl der Lichtquelle und die reflektierenden Objekte einer Szene beeinflusst werden. Auch die verwendeten Aufnahme- und Wiedergabesysteme spielen eine Rolle. Eine genauere Betrachtung folgt in Kapitel 4. Die Kombination aus diesen Faktoren

ergibt die spektrale Zusammensetzung des Lichts, welches durch unser Auge interpretiert wird. Auch das menschliche Auge gewichtet verschiedene Wellenlängen unterschiedlich (siehe Kapitel 2.3).

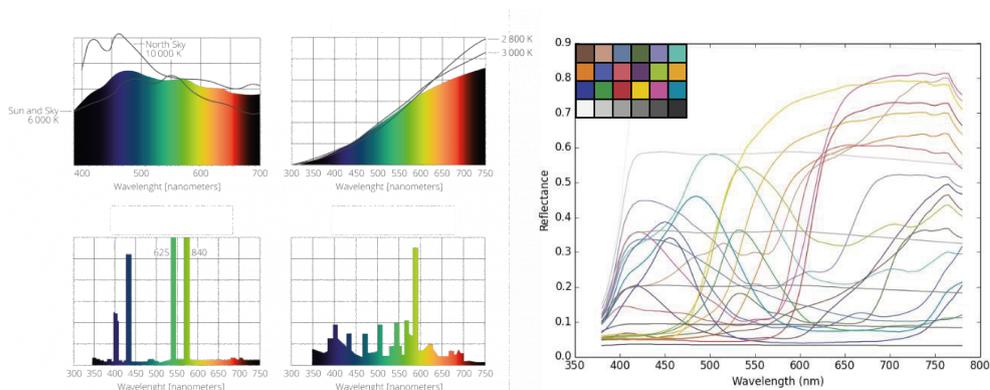


Abbildung 2.7.: Spektrale Eigenschaften verschiedener Lichtquellen (natürliches Licht, Glühlampen, Quecksilberdampfampe und Mehrdampfampe) und eines reflektierenden Colorcharts, frei nach [35], [24]

Abbildung 2.7 zeigt die unterschiedlichen spektralen Eigenschaften von verschiedenen Lichtquellen und die spektralen Reflektanzen eines Colorcharts.

Zunächst entscheidend ist die spektrale Zusammensetzung des Lichts. Jede Frequenz wird von den Objekten einer Szene unterschiedlich stark reflektiert bzw. absorbiert. Die Reflektanz eines Objekts ist also wellenlängenabhängig. Ein Farbchart reflektiert bestimmte Farbfrequenzen, die für Kameraproduktionen kritische Farbtöne beinhalten. Ihre spektrale Verteilung ist in der Abbildung zu sehen. Die Wahrnehmung durch das Auge ist ebenfalls frequenzabhängig. Bestimmte Farbtöne werden unterschiedlich stark wahrgenommen [36]. Eine genauere Betrachtung folgt in diesem Kapitel im Bereich Farbwahrnehmung.

2.2. Der Aufbau des menschlichen Auges

Abbildung 2.8 zeigt den Aufbau des menschlichen Auges und der Ganglienzellen innerhalb der Netzhaut. Das menschliche Auge nimmt starken Einfluss auf die Wahrnehmung des HVS.

Einfallendes Licht wird zunächst durch die Linse gebrochen und auf die Fovea entlang der Sichtachse fokussiert. Am Rande der Linse befinden sich Linsenmuskeln, die anders als bei starren Linsen einer Kamera nicht verschoben werden, sondern die Wölbung der Linse und damit den Brechungsgrad verändern [51] [44].

Über die Iris adaptiert das Auge die Helligkeiten des einfallenden Lichts. Die Iris-muskeln können je nach Intensität des einfallenden Lichts die Pupillenöffnung zwi-

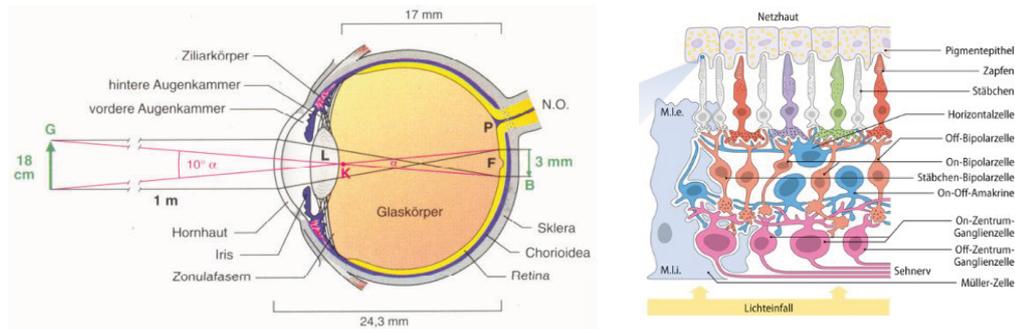


Abbildung 2.8.: Aufbau des menschlichen Auges und Ganglienzellen, frei nach [44] und [19]

schen 1,5 und 8mm variieren. Dadurch wirkt die Pupille wie eine Blende. Sie reagiert schnell innerhalb von 200 bis 500 Millisekunden [51] [44].

Auf der Netzhaut werden die Helligkeits- und Farbinformationen in Nervenreize übersetzt. Das menschliche Auge verfügt über ca. 120 Millionen Stäbchen, die besonders lichtempfindlich sind und daher vor allem für das Hell-Dunkel-Sehen bei schwachem Licht zuständig sind. Die ca. 6 Millionen Zapfen kommen für das Farbsehen und das Sehen bei hellem Licht zum Einsatz [51].

Auch innerhalb der Stäbchen und Zapfen kommt es zur **Helligkeitsadaption**. In dunklen Lichtsituationen werden die Stäbchen aktiviert und erhöhen ihre Empfindlichkeit. Dabei wird Rhodopsin als lichtempfindliches Pigment gebildet. Dieser Prozess nähert sich einer Sättigung an. Abbildung 2.9 zeigt den Verlauf der Dunkeladaption. Da bei Dunkelheit die Zapfen schneller adaptieren als die Stäbchen, sieht man vorerst weiter über die Zapfen. Mit dem Aufbau des Rhodopsin erreichen die Stäbchen nach kurzer Zeit eine höhere Empfindlichkeit als die Zapfen. Der Übergang vom Zapfensehen zum Stäbchensehen wird als Kohlrauschknick bezeichnet [25].

In hellen Lichtsituationen werden die Zapfen aktiviert während die Stäbchen weniger aktiv werden. Das gebildete Rhodopsin wird dabei in den Stäbchen wieder abgebaut und verhindert so eine Überbelichtung und daraus resultierende Blendung der Person. Die Anpassung an die Helligkeit geht entschieden schneller und dauert nur wenige Minuten. An den Grenzpunkten, an denen die Pupille voll geschlossen ist und das Rhodopsin abgebaut ist, oder der Helligkeitswechsel zu schnell eintritt, kommt es zur **Blendung** [25].

Die Retina verfügt über drei verschiedene Zapfentypen, die empfindlich auf unterschiedliche Wellenlängen reagieren. Wir unterscheiden zwischen L-Zapfen (Long), M-Zapfen (Medium) und S-Zapfen (Short), die auf lange, mittlere und kurze Wellenlängen reagieren. Je nach Farbtemperatur des Lichts, werden die Zapfen unterschiedlich stark angeregt und passen sich demnach unterschiedlich stark an die spektrale Zusammensetzung des Lichts an. Innerhalb der Zapfen wird Iodopsin in

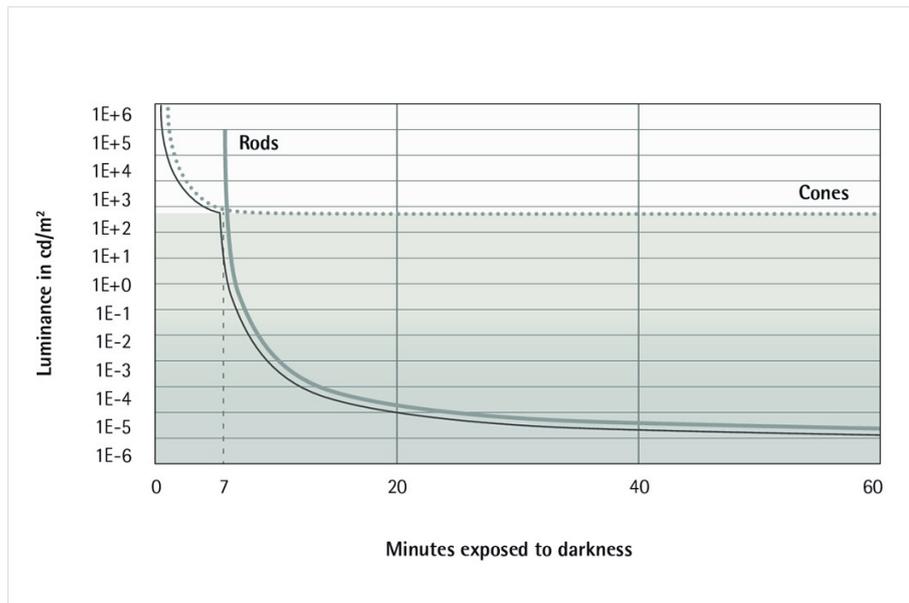


Abbildung 2.9.: Dunkeladaption des menschlichen Auges [25]

Abhängigkeit von der Wellenlänge abgebaut. Je nach Zusammensetzung des Lichts werden die Photopigmente der L-, M- und S-Zapfen unterschiedlich stark abgebaut. Die Adaption an die spektrale Zusammensetzung wird als **chromatische Adaption** bezeichnet [25].

2.3. Wahrnehmung des HVS

Aus den Wellenlängen der drei Zapfentypen ergibt sich der wahrnehmbare Spektralbereich, auf den in Kapitel 2.3.4 genauer eingegangen wird. Die Blendungsgrenze und die Empfindlichkeitsgrenze der Stäbchen bilden den wahrnehmbaren Kontrastumfang des HVS und werden im Folgenden in Kapitel 2.3.1 genauer betrachtet [44, S.2-5].

2.3.1. Helligkeits- und Kontrastwahrnehmung

Je nach Belichtungssituation sind in unserem visuellen System die Zapfen, Stäbchen oder beide kombiniert aktiv. In sehr hellen Belichtungssituationen, dem photopischen Sehen, nehmen wir die Umgebung nur durch unsere Zapfen wahr. Diesen Zustand haben wir bei Beleuchtungsichten von circa über 3 cd/m^2 und können bis zu einer Beleuchtungsichte von ca. 10^6 cd/m^2 adaptieren, bevor unsere Blendungsgrenze erreicht wird. Auch bis zu einer Beleuchtungsichte von $0,03 \text{ cd/m}^2$ sind die Zapfen noch aktiv, haben aber mit sinkender Helligkeit immer weniger

Einfluss auf unser wahrgenommenes Bild. Für den Bereich unter 3 cd/m^2 bis zu einer Leuchtdichte von 10^{-6} cd/m^2 nehmen wir Licht über die Stäbchen wahr. In den dunklen Helligkeitsbereichen ist dementsprechend das Nachtsehen durch die Stäbchen aktiv. Dieser Bereich wird als skotopisches Sehen bezeichnet. Die Zwischenbereiche zwischen $0,03 \text{ cd/m}^2$ und 3 cd/m^2 bilden den mesopischen Sichtbereich. Hier sind Stäbchen und Zapfen aktiv. In Abbildung 2.10 ist der wahrnehmbare Helligkeitsbereich visualisiert [20] [51, S. 30-32]. Andere Quellen berichten von einem Bereich von 10^{-6} cd/m^2 bis 10 cd/m^2 für das Stäbchensehen und einen Bereich von $0,01 \text{ cd/m}^2$ bis 10^8 cd/m^2 für das Zapfensehen [40].

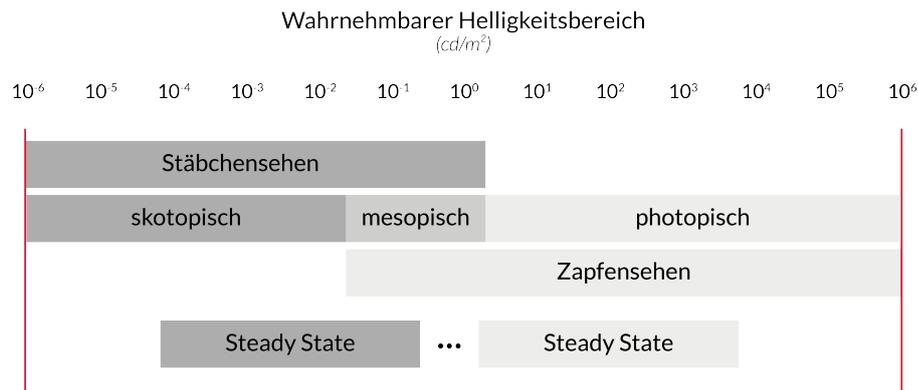


Abbildung 2.10.: Wahrnehmbarer Helligkeitsbereich frei nach [20] und [51]

Gehen wir von einem wahrnehmbaren Bereich von 10^{-6} bis 10^6 cd/m^2 aus, dann entspricht das einem Helligkeitsunterschied von 10^{12} . Berechnet man daraus die wahrnehmbaren Blendenstufen n für das HVS, so ergibt sich:

$$n = \log_2(10^{12}) = 12 \cdot \log_2(10) \approx 39.86 \quad (2.5)$$

Das menschliche Auge kann also theoretisch Helligkeitsunterschiede von knapp 40 Blendenstufen wahrnehmen. Das entspricht jedoch nur dem Bereich, den das menschliche Auge adaptieren kann. Der unadaptierte Bereich wird als steady-state bezeichnet und umfasst rund $3,7 \log$ Einheiten [40]. Das entspricht umgerechnet einem Kontrastumfang von ca. 1:5000 oder 12,3 Blendenstufen. Andere Studien sprechen von einem Steadystate von 2 bis 4 \log Einheiten. Es ist jedoch wichtig zu beachten, dass durch den Adaptionsprozess der Stäbchen und Zapfen, wie oben beschrieben, die Dauer der Stimulation entscheidenden Einfluss auf die Blendenstufen nimmt [40]. In dieser Arbeit wird von 12,3 Blendenstufen ausgegangen.

2.3.2. Nichtlinearität des HVS

Ein wichtiges Konzept für die Wahrnehmung von Helligkeitswerten ist die Nichtlinearität des HVS. Diese geht zurück auf die Leipziger Professoren Ernst Heinrich

Weber und Gustav Theodor Fechner. 1834 haben sie in verschiedenen Versuchen festgestellt, dass die Sinneswahrnehmung in einem nichtlinearen Verhältnis zu der Veränderung der Umgebung steht. Weber hat das nach ihm benannte Webersche Gesetz entwickelt. Es besagt, dass die Empfindung zwischen zwei Sinnesreizen von der Stärke des Ausgangsreizniveaus abhängig ist und konstant verläuft. Das könnte zum Beispiel eine Zeitung sein, die sowohl bei Sonnenlicht als auch bei Mondlicht gelesen werden kann. Der Kontrast zwischen den Buchstaben und dem weißen Papier bleibt dabei konstant. Fechner hat einen Zusammenhang vorgeschlagen, der die Empfindungsstärke mit der Reizstärke durch eine Logarithmusfunktion beschreibt. Stellt man die Formel von Weber nach der Empfindung E um, dann ergibt sich eine ähnliche Schreibweise wie für die Fechnerfunktion. Sie wurde daher zur Weber-Fechnerschen Funktion zusammengefasst [44, S. 39-40] [51, S. 32-35].

$$E = \frac{1}{c'} \cdot \ln(I) + C \quad (2.6)$$

Später wurde das Weber-Fechnersche Gesetz durch die Stevenssche Potenzfunktion abgelöst. Danach gilt:

$$E = k \cdot I^n \quad (2.7)$$

In der Contrast Sensitivity Function (CSF) hat Piet Gerard Joseph Barten 1999 formuliert, wie empfindlich das menschliche visuelle System bei verschiedenen räumlichen Frequenzen reagiert. Diese mathematische Funktion beruht auf den Modellen zur menschlichen Wahrnehmung und geht somit auch davon aus, dass die Beziehung zwischen der Helligkeit einer Szene und der wahrgenommenen Helligkeit nicht linear verläuft. Die von Barten entwickelte CSF zeigt, dass das menschliche visuelle System am empfindlichsten bei mittleren Raumfrequenzen reagiert und bei hohen und niedrigen Frequenzen abnimmt. Dabei hängt die Kontrastwahrnehmung auch von der Umgebungshelligkeit ab [6].

2.3.3. Rauschwahrnehmung

Die Sichtbarkeitsschwelle für Rauschen lässt sich je nach Quelle auf einen SNR von 42 Dezibel (dB) definieren. Bei dem Wert handelt es sich um einen linearen und unbewerteten Störabstand, der als durchschnittlicher Richtwert angesehen werden kann [51, S. 413-414]. Wie im vorherigen Kapitel zur Nichtlinearität des HVS schon angesprochen, verhält sich das visuelle System im Bezug auf die Wahrnehmung jedoch nicht linear. Auch die Rauschwahrnehmung des visuellen Systems hat dementsprechend keinen linearen Zusammenhang und reagiert unterschiedlich auf Raumfrequenzen. Um das visuelle Rauschen zu beschreiben, kann die Nichtlinearität des HVS simuliert werden, indem die CSF nach Barten im Frequenzspektrum des

Rauschsignals mit dem Rauschen verrechnet wird. Das Verfahren ist in der ISO-Norm ISO 15739 formuliert [54].



Abbildung 2.11.: Transformationsschritte zur Herleitung des visuellen Rauschen [4]

Abbildung 2.11 zeigt die Transformationsschritte, um das Rauschen eines Signals zu gewichten. Anstelle des SNR berechnet man einen Wert, der das visuelle Rauschen abbildet durch die Gewichtung über die CSF. Je höher der Rauschwert ist, desto stärker kann das Rauschen wahrgenommen werden. Rauschwerte, die den Wert 0 unterschreiten, sind für das menschliche Auge nicht sichtbar [4] [43].

Bei der Implementierung handelt es sich um einen relativ komplizierten Algorithmus mit vielen zu beachtenden Feinheiten, was den Rahmen dieser Arbeit überschreiten würde. In dieser Arbeit wird daher vereinfacht von einem sichtbaren Rauschen bei einem SNR von durchschnittlich 42 dB ausgegangen.

2.3.4. Farbwahrnehmung

Wie im Kapitel zum menschlichen Auge bereits angesprochen, sind die L-, M- und S-Zapfen für die Farbwahrnehmung verantwortlich. Die Zapfen reagieren auf Licht unterschiedlicher Wellenlängen ab einer Leuchtdichte von ca. 3 cd/m^2 , also im Bereich des mesopischen und photopischen Sehens. Die volle Empfindlichkeit der Zapfen erreicht das Auge bei einer Leuchtdichte von 10 cd/m^2 . Ab dort kann das volle Farbspektrum wahrgenommen werden. Mit den L-Zapfen nehmen wir den langwelligen Rotbereich, mit den M-Zapfen den mittelwelligen Grünbereich und mit den S-Zapfen den kurzwelligen Blaubereich wahr. Dabei reagieren die unterschiedlichen Zapfen jedoch nicht nur bei einer Wellenlänge, sondern bei einem Wellenlängenbereich, in dem diese sich unterscheiden. Dadurch ergibt sich ein wahrnehmbarer Spektralbereich von Wellenlängen zwischen 380nm und 780nm [44, S. 11-15] [51, S. 36-38].

Abbildung 2.12 zeigt die Empfindlichkeit der Zapfen für verschiedene Wellenlängen. Die Überschneidungen der Empfindlichkeiten der einzelnen Zapfen zeigen, wie die Farbmischung und damit Farbwahrnehmung beim menschlichen Auge funktioniert. Durch die Kombination der Zapfen entstehen die wahrgenommenen Zwischenfarben. Für ein oranges Farbchart werden beispielsweise die L- und M-Zapfen angeregt. Unser Gehirn macht aus der Kombination dieser Nervenreize die wahrgenommene orangene Farbe. Man spricht hierbei von additiver Farbmischung [44, S. 11-15] [51, S. 36-38].

Die Commission internationale de l'Éclairage (CIE) hat im Jahre 1931 ein Farbsystem zur Darstellung aller durch den Menschen wahrnehmbarer Farben erstellt.

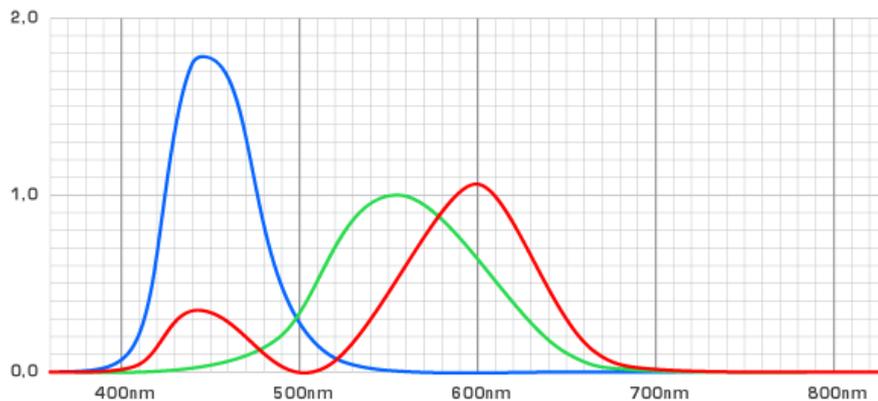


Abbildung 2.12.: Spektrale Empfindlichkeit des menschlichen Auges [3]

Das CIE-Normfarbsystem wurde auf Grundlage von drei Tristimuluswerten (X, Y, Z) entwickelt und basiert auf der Annahme, dass der Mensch Farben durch die Kombination aus drei Grundfarben wahrnimmt. Eine genaue Betrachtung des CIE-Normfarbsystems findet sich in Kapitel 3. Aus dem CIE-Normfarbsystem und den daraus resultierenden Farbräumen lassen sich auch die wahrnehmbaren Farbunterschiede ermitteln [36].

2.4. Sichtbarkeitsgrenzen des HVS

In der Einleitung wurde auf die Forschungsfragen eingegangen. Mit den Grundlagen zum HVS lässt sich die Frage, welche Anforderungen an Kamerasysteme auf Grundlage der Wahrnehmung gestellt werden können, wie folgt beantworten.

Das menschliche Auge nimmt Helligkeitsunterschiede im steady state von rund 12,3 Blendenstufen wahr. Auf Grundlage dessen lässt sich für Kamerasysteme somit definieren, dass der Sensor einen Dynamikumfang von 12,3 Blendenstufen abdecken muss. Es gilt zu beachten, dass diese Schlussfolgerung unter der Annahme getroffen wird, dass eine Adaption des Auges beim Abbilden des Kameramaterials nicht gewünscht ist. Für Sensoren, die diesen Richtwert überschreiten, stellt sich weitergehend die Frage, bis zu welcher Sensorvorverstärkung der Dynamikumfang der Kamera ausreicht, um die 12,3 Blendenstufen abbilden zu können.

Geht man von einer ungewichteten linearen Rauschwahrnehmung aus, dann liegt das sichtbare Rauschen bei einem Abstand von 42 dB. Unabhängig vom Ausspielmedium kann man für den szenenlinearen Bereich definieren, dass ein SNR von 42 dB der Grenze für ein qualitativ hochwertiges Signal entspricht. Für die Kamerasysteme lässt sich daraus die Anforderung definieren, bis zu welcher Vorverstärkung die Kamera einen Dynamikumfang von 42 dB erreicht.

Die Farbunterschiede der visuellen Wahrnehmung können über die Verwendung

des $L^*a^*b^*$ -Farbraums definiert werden. Die Grundlagen dazu werden in Kapitel 3 definiert und die Sichtbarkeitsgrenzen und Anforderungen an die Kamerasysteme für die Farbunterschiede des HVS abgeleitet.

3. High Dynamic Range und Wide Color Gamut

Bis zum Jahre 2015 war der SDR-Workflow bei einem BT.709 Farbraum der Standard im Broadcast. Mit der Standardisierung der HDR-Übertragungsfunktionen von PQ und HLG haben die Themenfelder HDR und WCG stetig größeren Einfluss auf den Broadcastbereich genommen [58]. Auch wenn bis heute der Großteil der Endgeräte die SDR-Wiedergabe als Standardwiedergabeformat hat, wird der Anteil an HDR-fähigen Endgeräten immer größer [12]. In diesem Kapitel werden die wichtigsten Grundlagen für die Themenfelder HDR und WCG erläutert und in einen Kontext zum bisherigen SDR-Standard gebracht. Dabei wird auf die erweiterten Möglichkeiten der Kontrastdarstellung und des größeren Farbraums eingegangen. Abbildung 3.1 zeigt die Übersicht über dieses Grundlagenkapitel.

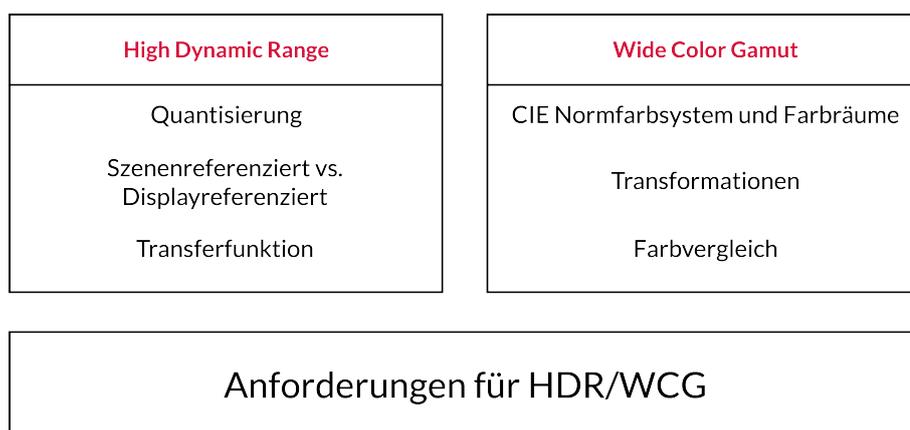


Abbildung 3.1.: Übersicht zum Grundlagenkapitel High Dynamic Range und Wide Color Gamut, frei nach [46]

3.1. High Dynamic Range

Unter HDR-Bildern versteht man hochkontrastige Bilder, die im Vergleich zu SDR-Bildern einen erhöhten Dynamikumfang aufweisen. Das kann sowohl in Richtung der Schwärzen durch einen niedrigeren Black Level, als auch in den hellen Bildbereichen durch eine erhöhte Spitzenleuchtdichte geschehen. HDR erweitert den Dynamikbereich, indem mehr Helligkeitsstufen abgebildet werden können. Dabei geht es nicht darum, die grundsätzliche Helligkeit des Bildes zu verändern, sondern mehr Detailzeichnung in den sehr hellen und sehr dunklen Bildbereichen zu ermöglichen und so dem tatsächlichen Kontrastumfang der menschlichen Wahrnehmung von 12,3 Blendenstufen näher zu kommen (vgl. Kapitel 2) [58].

3.1.1. Quantisierung

Wie in Kapitel 2 ausführlich beschrieben, ist die Wahrnehmung des Menschen nicht linear, sondern kann am ehesten durch einen logarithmischen Zusammenhang beschrieben werden. Kamerasensoren hingegen haben einen linearen Zusammenhang zur Beleuchtungsdichte des Sensors. Eine Verdopplung der Photonenmenge führt automatisch auch zu einer Verdopplung der Codewerte eines Pixels.

Geht man davon aus, dass gemäß der Weber-Konstanten das menschliche Auge Helligkeitsunterschiede von ca. 1% wahrnehmen kann, dann wären bei einer linearen Quantisierung der Codewerte Unterschiede dann sichtbar, wenn der Unterschied zwischen zwei benachbarten Codewerten mehr als 1% entspricht [22, S. 37]. Das gilt für alle Codewerte C für die gilt:

$$C \leq \frac{(C + 1) - C}{0.01} \quad (3.1)$$

Substituiert man in der Formel $(C + 1) - C = 1$, so zeigt sich, dass die Helligkeitsunterschiede für alle Codewerte kleiner 100 gilt. Bei einer 8 Bit Quantisierung, die standardmäßig für die Distribution von SDR-Content verwendet wird, entspräche das bei 256 Codewerten folgender Rechnung:

$$\text{f-stops} = \log_2 \left(\frac{256}{100} \right) \quad (3.2)$$

Mit einer linearen Zuordnung könnte man bei einer 8 Bit Quantisierung also nur 1,35 Blendenstufen Helligkeitswerte abbilden. Auch eine höhere Quantisierung von 10 Bit oder 12 Bit führt nur zu 3,36 und 5,36 Blendenstufen. Zudem würden die hellen Bildbereiche mit mehr Codevalues abgetastet werden als für die Wahrnehmung notwendig [22, S. 37]. Um diesem Phänomen entgegenzuwirken, werden die Codewerte über Quantisierungskurven, im folgenden auch Gammakurven genannt, bestimmten Helligkeitswerten zugeordnet.

3.1.2. Szenenreferenziert vs. Displayreferenziert

Bei der Zuordnung der logarithmisch wahrgenommenen Leuchtdichten zu bestimmten linearen Codewerten unterscheidet man zwei Ansätze:

Eine Codewertzuordnung, bei der die tatsächlichen physikalischen Leuchtdichten einer Szene erfasst werden, wird als szenenreferenziert bezeichnet. Dabei werden

die Leuchtdichten Codewerten zugeordnet, die in der realen Welt existieren, unabhängig davon, wie sie später dargestellt werden. Dadurch können mehr Helligkeitswerte erfasst werden als im Augenblick durch Wiedergabesysteme darstellbar sind [58].

Bei dem zweiten Ansatz werden die Codewerte bezogen auf ein bestimmtes Ausgabegerät angepasst. Dabei werden Codewerte speziell so zugeordnet, dass sie für ein Referenzdisplay korrekt dargestellt werden. Der zweite Ansatz wird als display-referenziert bezeichnet [58].

3.1.3. Transferfunktionen

Die Codewertzuordnung wird über Transferfunktionen beschrieben, also mathematische Funktionen, die den Zusammenhang zwischen den Leuchtdichten und den Codewerten beschreiben. Jeder Übertragungsstandard besteht aus zwei wesentlichen Transferfunktionen. Die Opto-Electronic Transfer Function (OETF) beschreibt die Umwandlung von den physikalischen Leuchtdichten in ein digitales Signal und ist in die Kamerasysteme implementiert. Um das digitale Signal für Displays abbildbar zu machen, muss das digitale Signal in Helligkeitswerte transformiert werden, die das Display anzeigen kann. Die Transferfunktion, die beschreibt, wie ein Display die digitalen Bilddaten in Lichtintensitäten umwandelt, ist durch die Electro-Optical Transfer Function (EOTF) beschrieben. Kombiniert man beide Transferfunktionen, so erhält man die kombinierte Transferfunktion, die als Opto-Optical Transfer Function (OOTF) bezeichnet wird. Für gängige SDR- und HDR-Standards wurden spezifische Transferfunktionen festgelegt. Abbildung 3.2 zeigt die Übertragungsstandards für SDR, HLG und PQ [59] [61] [60] [58] [45].

Die Grafik unterscheidet zwischen der szenenlinearen und displaylinearen Domäne. Diese beziehen sich auf die mathematischen Zusammenhänge zwischen den Helligkeitswerten und den Signalwerten. Ein Bild befindet sich in der szenenlinearen Domäne, wenn die Signalwerte linear proportional zu den tatsächlichen Lichtintensitäten einer Szene sind. Als displaylineare Domäne bezeichnet man entsprechend Bilder, deren Signalwerte linear proportional zu den Lichtintensitäten eines Referenzdisplays sind. Die verschiedenen Standards wurden zu verschiedenen Zeiten mit verschiedenen Intentionen implementiert.

Der ITU Standard „RECOMMENDATION ITU-R BT.709“ wurde im Jahr 1990 eingeführt und seitdem mehrmals aktualisiert. Er spezifiziert verschiedene Parameter für den HDTV-Standard. Darunter fällt unter anderem die OETF für die Transformation vom szenenlinearen Bild in ein elektrisches Signal [59]. Die EOTF zur Transformation des elektrischen Signals in ein displaylineares Bild wurde in der „RECOMMENDATION ITU-R BT.1886“ spezifiziert [61]. Die Transferfunktionen orientieren sich an dem Stevens Law und wurden über einen Gammawert von 2,2 implementiert [55] [59]. In den Schwarzen unterhalb eines Signalwerts von 0.018 ist die OETF

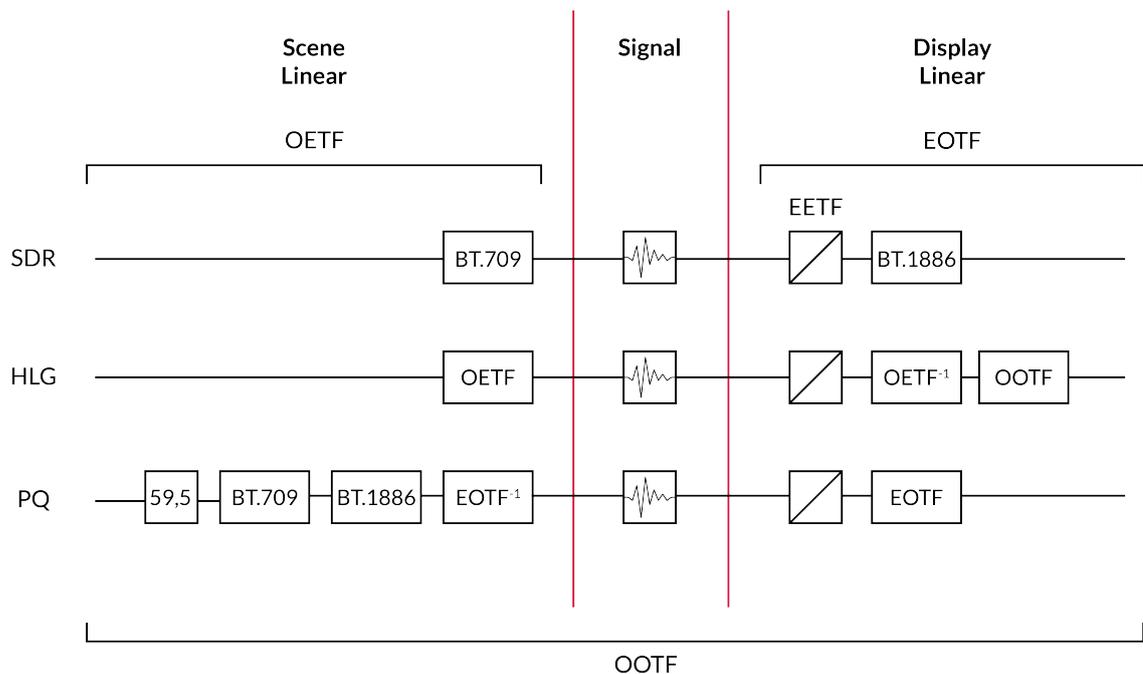


Abbildung 3.2.: Schematische Darstellung der Transferfunktionen der SDR, HLG und PQ Übertragungsstandards (freie Darstellung aus persönlichem Gespräch mit Justus Mai und nach [59] [61] [60] [58])

mit einem konstanten Skalierungsfaktor von 4,5 implementiert. Dadurch lässt sich die Quantisierung in den Schwärzen limitieren, um das Auslese- und Photonenrauschen in den sehr dunklen Bildbereichen zu reduzieren. In der ITU-R BT.1886 wurde die Quantisierungslimitierung später durch ein Offset in dem Schwarzwert implementiert [22, S. 38]. Die genaue Implementierung findet sich in den jeweiligen ITU-Standards [59] [61].

Für die Implementierung der HDR-Standards gibt es mit HLG und PQ zwei Ansätze, die beide in der „RECOMMENDATION ITU-R BT.2100“ standardisiert sind [58].

HLG wurde durch die British Broadcasting Company (BBC) in Kooperation mit dem japanischen Unternehmen Nippon Hōsō Kyōkai (NHK) entwickelt und erstmals 2015 von der Association of Radio Industries and Businesses (Arib) im Standard STD-B67 standardisiert, bevor sie in den ITU-R BT.2100 Standard aufgenommen wurden [5]. Bei dem HLG-Standard handelt es sich um einen displayreferenzierten Ansatz, entsprechend sind die Codewerte wie bei der „ITU-R BT.709“ auf das Referenzdisplay optimiert. Ein wichtiger Aspekt für die Entwicklung des HLG-Standards war die Rückwärtskompatibilität mit dem bisherigen SDR-Standard. Die Transferfunktion verläuft zum großen Teil annähernd wie die SDR-Transferfunktion, in den Highlights, jedoch durch einen stärkeren logarithmischen Verlauf mit einem Systemgamma von 1,2. Dadurch kann der HLG-Standard ohne zusätzliche Metadaten in bestehende SDR-Systeme transformiert werden. Die Highlightkompression erlaubt es, eine höhere Spitzenleuchtdichte von bis zu 1000 cd/m² darzustellen. Der

Schwarz- und Weißlevel des HLG-Standards kann auf das jeweilige Referenzdisplay optimiert werden, da in dem Standard der Peak Weißpunkt und der Schwarzwert angepasst werden können [58]. Die vollständige Implementierung findet sich im „ITU-BT2100“ Standard [58].

Im Gegensatz zum displayreferenzierten Ansatz von HLG ist PQ als szenenreferenzierter Ansatz implementiert. Er wurde durch die Dolby Laboratories Inc. und die Society of Motion Pictures and Television Engineers (SMPTE) entwickelt und 2014 im SMPTE Standard ST2084 standardisiert [45]. Auch der PQ-Standard wurde später in der „ITU-R BT.2100“ mit dem HLG-Standard zusammengefasst [58]. Der PQ Standard wurde auf Grundlage des Barten-Modells entwickelt und orientiert sich sehr nah an der Wahrnehmungsschwelle des HVS. Dadurch können Quantisierungsverluste durch die Implementierung minimiert werden, wodurch sich sehr große Dynamikbereiche abbilden lassen. Der PQ-Standard ist für einen Helligkeitsbereich von 0.005 bis 10000 cd/m² optimiert. Damit kann der Standard weit mehr Helligkeitswerte darstellen als aktuelle Wiedergabegeräte und der HLG-Standard abbilden können [58].

Sowohl die HLG- als auch die PQ-Pipeline beinhaltet eine OOTF. Diese wird auch als „Reference OOTF“ bezeichnet. Die Beleuchtung des Displays und die Leuchtdichte des Lichts, welche auf einen Sensor fällt, verläuft nicht linear. Die Differenz wird über die OOTF kompensiert. Im SDR-Workflow ist diese Kompensation in den BT.709 und BT.1886 implementiert. Für PQ wird diese Signaltransformation deshalb auf den Helligkeitsbereich des Standards normiert über den Faktor 59,5208 und anschließend die Differenz mit der selben Transformation des SDR-Standards kompensiert [58].

3.2. Wide Color Gamut

Als WCG versteht man Farbräume, die ein erweitertes Farbspektrum abbilden können. Farbräume bezeichnen den Bereich an Farben, der durch ein System abgebildet werden kann. In diesem Kapitel sollen die Farbräume und die Eigenschaften von WCG im Bezug auf das bisherige SCG erläutert werden.

3.2.1. CIE-1931-Normfarbsystem

Wie in Kapitel 2 bereits angedeutet, hat die CIE im Jahre 1931 ein Farbsystem zur Darstellung aller durch den Menschen wahrnehmbarer Farben erstellt. Das Farbsystem entstand aus der Annahme, dass jede wahrnehmbare Farbe durch eine Kombination von den 3 Primärfarben X, Y und Z additiv gemischt werden kann. Dazu wurden Experimente zur Farbwahrnehmung durchgeführt. Die Probanden

haben ein Licht vordefinierter Wellenlänge in einem 2° Sichtfeld gezeigt bekommen, welches sie aus drei Lasern der 3 Primärfarben additiv mischen sollten. Die Primärfarben sind dabei als Laser der Wellenlänge 700nm, 546.1nm und 435.8nm definiert, welche mit der Wahrnehmung von Rot, Grün und Blau korrelieren. Später wurde dasselbe Experiment mit einem Betrachtungswinkel von 10° durchgeführt um ein größeres Sichtfeld zu simulieren. Die beiden Messkurven werden als 2°-Standardbeobachter und 10°-Standardbeobachter beschrieben [36] [11].

Um die Farbe unabhängig von der Helligkeit beschreiben zu können, wurde das CIE-1931 Chromatizitätsdiagramm entwickelt, welches jede wahrnehmbare Farbe durch zwei Farbwerte x und y sowie einen Helligkeitswert Y beschreiben kann. Die Berechnung der Farbwerte erfolgt durch die folgende Formel, wobei der Helligkeitswert Y dem Tristimuluswert Y entspricht [36].

$$x = \frac{X}{X + Y + Z} \quad (3.3)$$

$$y = \frac{Y}{X + Y + Z} \quad (3.4)$$

Durch die x und y Position kann jede wahrnehmbare Farbe als Farbkoordinate dargestellt werden. Mappt man alle Farben auf eine zweidimensionale Darstellung, erhält man ein hufeisenförmiges Diagramm. Dieses wurde durch die CIE als CIE-1931 Chromatizitätsdiagramm definiert. Innerhalb des Diagramms befinden sich alle Farben, wobei die Farben nah an den Rändern die gesättigten Farben abbilden und entsättigter werden, je näher sie sich der Mitte annähern. Im Zentrum ist der Weißpunkt des Diagramms definiert.

Abbildung 3.3 zeigt das Chromatizitätsdiagramm mit verschiedenen integrierten Farbräumen. Im Folgenden soll auf die wichtigsten Farbräume eingegangen werden.

3.2.2. Farbräume

Bei Farbräumen handelt es sich um eine mathematische Zuordnung von Farbkoordinaten, die durch verschiedene Bezugspunkte definiert wird. Die Bezugspunkte werden als Primärvalenzen bezeichnet und definieren, aus welchen Primärfarbwerten die Farben in dem Bezugssystem zusammengesetzt werden. Es gibt eine Vielzahl an verschiedenen Farbräumen, die jeweils aus unterschiedlichen Gründen entwickelt wurden. Grundsätzlich kann man sagen, dass Farbräume immer in Abhängigkeit von dem Ausspielmedium, für die sie verwendet werden, entwickelt wurden. Beispielsweise also für ein Referenzdisplay, welches die Farben des Referenzfarbraums „ITU-R BT.709“ abbilden kann, oder ein Drucker der aus den CMYK-

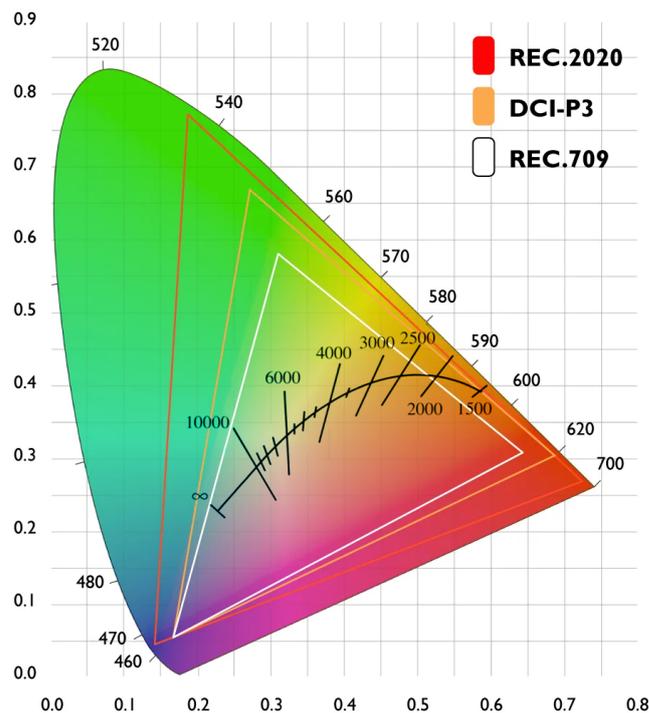


Abbildung 3.3.: Das CIE-1931 Chromatizitätsdiagramm mit Farbräumen, frei nach [7], [1]

Farbtönen die Farbtöne auf dem Bild zusammensetzt. Kein heutiges technisches Gerät kann den gesamten Farbraum abbilden, der von dem HVS wahrgenommen werden kann.

Der bisherige Standardfarbraum für SCG-Produktionen ist in der „ITU-R BT.709“ definiert. Dabei handelt es sich um einen RGB-Farbraum mit den Primärvalenzen $x = 0,64$ und $y = 0,33$ für Rot, $x = 0,30$ und $y = 0,60$ für Grün und $x = 0,15$ und $y = 0,06$ für Blau. Der Farbraum kann in 8 oder 10 Bit Quantisierung abgebildet werden, was ca. 16,8 Millionen bzw. 1,07 Milliarden Farbabstufungen entspricht [59]. Der Farbraum deckt je nach Quelle rund 35,9% des Bereiches an sichtbaren Farben ab, die für das HVS sichtbar sind [53] [64].

Bei WCG-Produktionen wird meistens der ITU-R BT.2020 Standard verwendet. Dabei handelt es sich um einen RGB-Farbraum, der ein deutlich größeres Spektrum an Farben abbilden kann. Die Primärvalenzen liegen bei $x = 0,708$ und $y = 0,292$ für Rot, bei $x = 0,170$ und $0,797$ für Grün und bei $x = 0,131$ und $0,046$ für Blau. Der Farbraum kann in 10 und 12 Bit Quantisierung abgebildet werden, was 1,07 Milliarden bzw. 68,72 Milliarden Farbabstufungen entspricht [60]. Der Farbraum deckt rund 75,8% des Bereiches an sichtbaren Farben ab und damit doppelt so viele Farben wie der „ITU-R BT.709“ Farbraum [53].

Viele Endgeräte können die Anforderungen des ITU-R BT.2020 Standards noch

nicht vollständig abdecken, weshalb viele auf den aktuellen Kinostandard DCI P3 zurückgreifen. Er wurde ursprünglich für die Filmprojektion in der Filmindustrie und in Heimkinos entwickelt und deckt rund 45,5% der wahrnehmbaren Farben ab. Er ist in dem RP 431-2 Standard der SMPTE definiert und liefert eine gute Übergangstechnologie auf dem Weg zur Ausspielung mit BT.2020 [52].

Neben den hier genannten Farbräumen gibt es eine Vielzahl an anderen Farbräumen, die unterschiedliche Anwendungsfälle spezifizieren. Die für diese Arbeit wichtigen sind jedoch der BT.709 und BT.2020 Farbraum.

3.2.3. Farbraumtransformation

Um Bilddaten von einem Farbraum in einen anderen Farbraum zu transformieren, müssen die RGB-Farbwerte des Quellfarbraums in die Tristimuluswerte des CIE-Normfarbsystems transformiert werden. Die Transformation zwischen den Farbräumen lässt sich über Transformationsmatrizen realisieren. Das CIE-Normfarbsystem dient als Bezugsfarbraum. Aus ihm kann dann durch eine zweite Matrizenmultiplikation in den Zielfarbraum transformiert werden.

Farbraumtransformation durch Matrizenmultiplikation



Abbildung 3.4.: Farbraumtransformation über das CIE-Normfarbsystem

Abbildung 3.4 zeigt die Farbraumtransformation. Jeder Farbraum hat eine spezifische Transformationsmatrix. Die RGB/XYZ Matrizen sind nicht immer bekannt, lassen sich jedoch über den Referenzweißpunkt und die x und y Werte der Primärvalenzen berechnen. Der folgende Code zeigt eine Implementierung der RGB/XYZ Matrizen nach den mathematischen Ausführungen von Bruce Lindbloom [41].

```

1 import numpy as np
2
3 def calc_ConvMatrix(xr, yr, xg, yg, xb, yb, xw, yw, Yw):
4     Xw = (xw * Yw) / yw
5     Yw = Yw
6     Zw = ((1 - xw - yw) * Yw) / yw
7
8     Xr = xr / yr
9     Yr = 1
10    Zr = (1 - xr - yr) / yr
11    Xg = xg / yg
12    Yg = 1
13    Zg = (1 - xg - yg) / yg
14    Xb = xb / yb
15    Yb = 1
16    Zb = (1 - xb - yb) / yb
17
  
```

```

18     XYZ_White = np.array([[Xw], [Yw], [Zw]])
19     XYZ_Matrix = np.array([[Xr, Xg, Xb], [Yr, Yg, Yb], [Zr, Zg, Zb
20     ]])
21     SrSgSb = np.dot(np.linalg.inv(XYZ_Matrix), XYZ_White)
22     Sr = SrSgSb[0]
23     Sg = SrSgSb[1]
24     Sb = SrSgSb[2]
25
26     M = np.array([[Sr * Xr), (Sg * Xg), (Sb * Xb)], [(Sr * Yr), (
27     Sg * Yg), (Sb * Yb)], [(Sr * Zr), (Sg * Zg), (Sb * Zb)]]
28     M = M[:, :, 0]
29     return M

```

Listing 3.1: Berechnung der RGB/XYZ Matrizen durch die Primärvalenzen

Da es sich bei der Transformation über das CIE-Normfarbsystem um zwei Matrizenmultiplikationen handelt, kann dieser Schritt in eine kombinierte Multiplikation zusammengefasst werden.

3.2.4. Weißpunkttransformation

Wie in Abbildung 3.3 zu sehen, sind alle Farbräume einem bestimmten Weißpunkt zugeordnet. Durch die chromatische Adaption des Auges wird bei unterschiedlichen Farbtemperaturen ein unterschiedlicher Farbton als Weiß wahrgenommen (vgl. Kapitel 2). Farbräume werden einem vordefinierten Referenzweißpunkt zugeordnet, der die Anwendung des Farbraums am besten widerspiegelt. Für Fernsehstandards wie die ITU-R BT.709 und die ITU-R BT.2020 entspricht das dem D65 Weißpunkt, bei einer Farbtemperatur von 6504 Kelvin [59] [60]. Diese Farbtemperatur entspricht dem durchschnittlichen Tageslicht. Andere Standards mit anderen Anwendungszwecken verwenden oft auch unterschiedliche Weißpunkte. Die Druckindustrie arbeitet beispielsweise oft mit dem D50 Weißpunkt, der einer Farbtemperatur von 5000 Kelvin zugeordnet ist. Der Weißpunkt beeinflusst stark, wie Farben in einem Farbraum wahrgenommen werden. [41]

Um einen Weißpunkt in einen anderen zu transformieren, nutzt man chromatische Adaptionalgorithmen wie beispielsweise das XYZ-Scaling, die Bradford-Methode oder die Von Kries-Methode. Dabei handelt es sich um eine lineare Transformation bei der die Quellfarbe (X_Q, Y_Q, Z_Q) durch eine Matrizenmultiplikation in eine Zielfarbe (X_Z, Y_Z, Z_Z) transformiert wird [41]. Es gilt:

$$\begin{bmatrix} X_Q \\ Y_Q \\ Z_Q \end{bmatrix} = [M] \begin{bmatrix} X_Z \\ Y_Z \\ Z_Z \end{bmatrix} \quad (3.5)$$

Die Transformationsmatrix ermittelt sich in drei Schritten. Zunächst wird der XYZ-Wert in die Cone Response Domäne (ρ, γ, β) transformiert. Anschließend werden die Vektorkomponenten abhängig von der jeweiligen Adaptionsmethode mit einer Faktormatrix multipliziert. Abschließend wird die Cone Response Domäne zurücktransformiert in den XYZ Bereich [41]. Es gilt:

$$[M] = [M_A]^{-1} \begin{bmatrix} \rho_D/\rho_S & 0 & 0 \\ 0 & \gamma_D/\gamma_S & 0 \\ 0 & 0 & \beta_D/\beta_S \end{bmatrix} [M_A] \quad (3.6)$$

$$\begin{bmatrix} \rho_S \\ \gamma_S \\ \beta_S \end{bmatrix} = [M_A] \begin{bmatrix} X_{WS} \\ Y_{WS} \\ Z_{WS} \end{bmatrix} \quad (3.7)$$

$$\begin{bmatrix} \rho_D \\ \gamma_D \\ \beta_D \end{bmatrix} = [M_A] \begin{bmatrix} X_{WD} \\ Y_{WD} \\ Z_{WD} \end{bmatrix} \quad (3.8)$$

In dieser Arbeit wird die Transformation über die Bradford-Methode verwendet. Für die gilt folgende Transformationsmatrix [41]:

Method	Bradford-Matrix		
Bradford	0.895100	0.266400	-0.161400
	-0.750200	1.713500	0.036700
	0.038900	-0.068500	1.029600

Tabelle 3.1.: Bradford-Matrix

3.2.5. Vergleich von Farbwerten

Auch wenn das CIE-Normfarbsystem jede wahrnehmbare Farbe durch zwei Koordinaten beschreiben kann, die theoretisch miteinander verglichen werden können, hat das System dennoch einige Einschränkungen. Eine entscheidende für den Vergleich von Farben ist, dass die Farbabstände nicht perzeptuell gleichabständig sind. Das bedeutet, dass gleiche Unterschiede in den Abständen nicht unbedingt bedeuten müssen, dass ein Betrachter die Unterschiede gleichabständig wahrnimmt. Aus diesem Grund wurde 1976 der CIE-LAB Farbraum eingeführt. Dieser ist näher an

der menschlichen Wahrnehmung, da Farbunterschiede in ihm besser abgebildet werden als in dem CIE-Normfarbsystem. Im gleichen Zuge wurde auch der CIE-LUV Farbraum eingeführt. Dieser ist aber eher für den Vergleich von Beleuchtungen entwickelt worden und wird in dieser Arbeit deshalb nicht angewendet [36].

Um Farbwerte beispielsweise von Kamerasystemen zu vergleichen, müssen die Pixelwerte in die Tristimuluswerte (X, Y, Z) transformiert werden. Aus den Tristimuluswerten lassen sich die Werte wiederum in den $L^*a^*b^*$ -Farbraum transformieren. L gibt dabei die Helligkeit der Farbe zwischen 0 und 100 an. a^* beschreibt die Farbtiefe auf einer Skala mit Signalwerten zwischen -128 (grün) bis 127 (rot). b^* beschreibt die Farbtiefe auf einer Skala mit Signalwerten zwischen -128 (blau) bis 127 (rot) [36].

Im $L^*a^*b^*$ -Farbraum sind die Farbabstände perzeptuell gleichabständig. Dadurch können wahrgenommene Farbabstände über den euklidischen Abstand zwischen zwei Farbpositionen ermittelt werden. Die einfachste Methode ist die Delta E Berechnung, die 1976 von der CIE definiert wurde [36]. Sie berechnet sich wie folgt:

$$\Delta E_{76} = \sqrt{(L_2^* - L_1^*)^2 + (a_2^* - a_1^*)^2 + (b_2^* - b_1^*)^2} \quad (3.9)$$

1994 und 2000 wurden zwei weitere Berechnungsmodelle vorgestellt, in denen bestimmte Wahrnehmungsverzerrungen durch die Helligkeit, Chroma und Farbtöne berücksichtigt werden. Eine genaue Beschreibung dieser beiden Standards findet sich in der CIE Technical Report von 2004 [36]. Je nach Umgebungshelligkeit, Größe der Probe, Art der Präsentation (Auspielmedium), Erfahrung, Alter des Betrachters und die Zeit für die Adaption lassen sich Fehler unterschiedlich wahrnehmen. Die Grenze der Sichtbarkeit liegt bei einem ΔE von etwa 1 [63].

3.3. Vorgaben an Kamerasysteme

In der Einleitung wurde die Forschungsfrage definiert: „Welche spezifischen Anforderungen stellen aktuelle HDR-WCG Standards an die Kamerasysteme?“ Mit den erläuterten Grundlagen in diesem Kapitel können die Anforderungen definiert werden.

Da das menschliche Auge einen Farbunterschied von ΔE unter 1 im $L^*a^*b^*$ Farbraum nicht unterscheiden kann, sollte die Abweichung einer Kamera zum Referenzfarbraum oder zu Kamerasystemen im selben Referenzfarbraum nicht unter 1 ΔE fallen. Diese Anforderung setzt voraus, dass bei der Kameracharakterisierung keine Farbtransformationen oder Farbshifts verwendet werden, die der Kamera einen spezifischen Look verpassen. Kameras, die gemeinsam im selben Setup verwendet

werden, sollten keine Abweichungen über $1 \Delta E$ haben, um die Kameras unterhalb der Sichtbarkeitsgrenze zu matchen.

Um die Anforderungen an Kamerasysteme zu definieren, kann ermittelt werden, wie viele Blendenstufen der Kamerasensor abbilden muss, um die SDR- und HDR-Standards voll auszureizen. Zudem kann definiert werden, welche Vorgaben sich aus den Anforderungen an aktuelle Referenzdisplays ergeben.

Um die Blendenanforderungen an den Kamerasensor zu ermitteln, kann definiert werden, welcher Referenzweißpunkt und Referenzschwarzpunkt in der displaylinearen Domäne benötigt wird. Durch die Transformation des Referenzweißpunkts und des Referenzschwarzpunkts in die szenenlineare Domäne können dort die benötigten Blendenstufen über folgende Formel ermittelt werden:

$$f - stops = \log_2\left(\frac{W_{reference}}{B_{reference}}\right) \quad (3.10)$$

Dadurch lässt sich die Anzahl der Blendenstufen ermitteln, die benötigt werden, um bestimmte Luminanzwerte in der displaylinearen Domäne abzubilden.

In der „ITU-R BT.709“ ist der Referenzweißpunkt, den ein SDR-Display erreichen sollte, auf 100 cd/m^2 definiert [59]. Der Referenzschwarzpunkt wird im „ITU-R BT.1886“ auf einen Wert zwischen 0.1 für einen moderaten Blacklevel und 0.01 für einen tiefen Blacklevel definiert [61]. In dieser Arbeit wird daher von einem Blacklevel von 0.05 ausgegangen. Der benötigte Dynamikumfang wurde durch den folgenden Quellcode ermittelt.

```
1 ref_white = colour.models.oetf_inverse_BT709(colour.models.
    eotf_inverse_BT1886(100, 0, 100))
2 ref_black = colour.models.oetf_inverse_BT709(colour.models.
    eotf_inverse_BT1886(0.05, 0, 100))
3 f_stops_sdr = np.log2(ref_white / ref_black)
```

Listing 3.2: Berechnung des benötigten Dynamikumfangs für den SDR-Workflow

Für den SDR-Workflow ergibt sich dadurch ein benötigter Dynamikumfang von 6,74 Blendenstufen.

Für den HLG-Workflow ist in der „ITU-R BT.2100“ eine nominale Spitzenleuchtdichte des Displays von 1000 cd/m^2 definiert. Die minimale Leuchtdichte des Displays sollte gemäß des Standards größer als 0.005 cd/m^2 sein. Für diese Arbeit wird der Wertebereich von $0.005 - 1000 \text{ cd/m}^2$ angenommen [58]. Der benötigte Dynamikumfang lässt sich wie folgt berechnen:

```
1 ref_white = colour.models.oetf_inverse_BT2100_HLG(1000)
2 ref_black = colour.models.oetf_inverse_BT2100_HLG(0.005)
3 f_stops_hlg = np.log2(ref_white / ref_black)
```

Listing 3.3: Berechnung des benötigten Dynamikumfangs für den HLG-Workflow

Für den HLG-Workflow ergibt sich dadurch ein benötigter Dynamikumfang von 14,67 Blendenstufen.

Für den PQ-Workflow ist in der ITU-R BT.2100 eine maximale Spitzenleuchtdichte des Displays von 10000 cd/m² definiert. Die minimale Leuchtdichte des Displays sollte gemäß des Standards ebenfalls 0.005 cd/m² sein, deshalb wird auch hier 0.005 cd/m² als Referenzschwarzpunkt verwendet [58]. Der benötigte Dynamikumfang wird für den PQ-Workflow für drei Referenzweißpunkte bei 1000 cd/m², 4000 cd/m² und 10000 cd/m² ermittelt. Der folgende Code wird dafür verwendet:

```
1 ref_white_10000 = colour.models.ootf_inverse_BT2100_PQ(10000)
2 ref_white_4000 = colour.models.ootf_inverse_BT2100_PQ(4000)
3 ref_white_1000 = colour.models.ootf_inverse_BT2100_PQ(1000)
4 ref_black = colour.models.ootf_inverse_BT2100_PQ(0.005)
5
6 f_stops_PQ_10000 = np.log2(ref_white_10000 / ref_black)
7 f_stops_PQ_4000 = np.log2(ref_white_4000 / ref_black)
8 f_stops_PQ_1000 = np.log2(ref_white_1000 / ref_black)
```

Listing 3.4: Berechnung des benötigten Dynamikumfangs für den PQ-Workflow

Für den PQ-Workflow ergibt sich dadurch ein benötigter Dynamikumfang von 11,02 Blendenstufen bei 1000 cd/m², 12,82 Blendenstufen bei 4000 cd/m² und 14,02 Blendenstufen bei 10000 cd/m².

4. Kameratechnik

Kamerasysteme setzen sich grundsätzlich aus drei Kernelementen zusammen: der Optik, dem Bildwandler und der Verarbeitungseinheit. Der Bildwandler und die Verarbeitungseinheit lassen sich zu den elektrischen Kameraelementen zusammenfassen [34]. In diesem Grundlagenkapitel soll kurz auf alle Kameraelemente eingegangen und anschließend die Eigenschaften der Kameraelemente erläutert werden. Ein besonderer Fokus liegt dabei auf den potentiellen Abbildungsfehlern, die im Prozess der Bildgebung von der Szene zum fertigen Bild entstehen können. Anschließend wird auf die Qualitätskriterien von Kamerasystemen eingegangen, anhand derer man Kamerasysteme bewerten und vergleichen kann. Zuletzt werden alle Kamerasysteme und Messinstrumente erklärt, die in den Messungen in Kapitel 5 und 6 verwendet werden.

4.1. Eigenschaften elektrischer Kameraelemente

4.1.1. Bildwandler

Bildwandler dienen in Kamerasystemen der Umwandlung von Lichtphotonen in elektrische Signale. Die Sensorfläche besteht aus Halbleitermaterial, aus dem durch den inneren photoelektrischen Effekt Elektronen herausgelöst werden (vgl. Kapitel 2) [51, S. 390]. Je effizienter Kamerasensoren die Photonen in elektrische Signale umwandeln, desto höher ist ihre Gesamtquanteneffizienz. Die Formel in Abhängigkeit von der Wellenlänge λ lautet:

$$\eta(\lambda) = \frac{\mu_e}{\mu_p} \quad (4.1)$$

μ_e beschreibt dabei die Anzahl an Elektronen, die durch das eintreffende Licht auf dem Sensor erzeugt werden, und μ_p beschreibt die Anzahl an Photonen, die den Sensor bestrahlen [18, S. 7-8].

Kamerasysteme gibt es mit einer Vielzahl an verschiedenen Sensorarten:

In frühen Fernsehkameras kamen Röhrenbildwandler zum Einsatz. Röhrenbildwandler bestehen aus einer lichtempfindlichen Schicht, einer Halbleiterschicht und einer

Elektronenstrahlröhre. Die lichtempfindliche Schicht der Röhrenbildwandler wird von den Lichtphotonen getroffen und setzt dabei Elektronen frei. Diese werden auf einer Halbleiterplatte aufgefangen und speichern somit die Lichtverteilung der Szene. Durch den Elektronenstrahl wird die Halbleiterplatte Zeile für Zeile abgetastet, um das gespeicherte Bild aufzulösen. Dabei entsteht ein elektrisches Signal, das proportional zur gespeicherten Ladung und damit proportional zur Helligkeit der entsprechenden Bildpunkte ist [51, S. 408-411].

In modernen Kamerasystemen gibt es eine Vielzahl an verschiedenen Sensorarten. Zu den bekanntesten modernen Bildwandlern zählen die Charge Coupled Device (CCD) und Complementary Metall Oxid Semiconductor (CMOS) Sensoren, die in vielen Kameras verbaut sind.

Ein klassischer CCD-Sensor besteht aus einer Reihe von Metall Oxid Semiconductor (MOS)-Kondensatoren, die in Pixeln angeordnet sind und als lichtempfindliche Elemente dienen. Der Sensor besteht aus einer Oxidschicht, die eine Elektrode von dem Halbleiter trennt. Durch eine angelegte Spannung an der Elektrode des MOS-Kondensators bildet sich eine Potentialsenke, in der Elektronen gesammelt werden können. Abbildung 4.1 verdeutlicht den Aufbau der CCD-Zellen [51, S. 399-407].

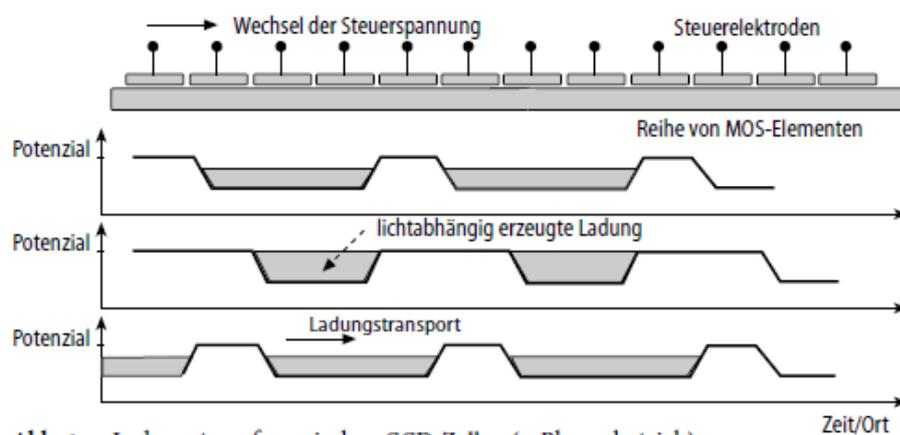


Abb. 6.13. Ladungstransfer zwischen CCD-Zellen (4-Phasenbetrieb)

Abbildung 4.1.: Ladungstransfer von CCD-Zellen [51, Abb. 6.13]

Das Bildaufnahmeverfahren bei CCD-Sensoren verläuft in drei Schritten. Im ersten Schritt wird das einfallende Licht durch den photoelektrischen Effekt in Elektronen-Loch-Paare umgeformt. Die Elektronen sammeln sich in der Potentialsenke, während die positiven Löcher in das Innere des Halbleiters gedrängt werden. Die gesammelten Elektronen sind dabei proportional zu den Photonen, die auf den Pixel fallen. Im zweiten Schritt kommt es zum Ausleseverfahren. Wie der Name Charge Coupled Device (CCD) schon zeigt, wird das Ausleseverfahren über den Ladungsaustausch zwischen einzelnen Pixeln realisiert. Die einzelnen MOS-Kondensatoren sind aneinandergereiht. Durch eine Verschiebung des Spannungsniveaus auf den Nachbapixel werden die Elektronen verschoben. Durch eine zyklische Wiederholung der Verschiebung des Spannungsniveaus können Pixel nacheinander ausgelesen werden. Das ausgelesene Signal wird im letzten Schritt verstärkt und über den

Analog-Digital-Wandler digitalisiert. Die daraus resultierenden Signalwerte bilden die Rohdaten des Kamerasensors [51, S. 399-407].

Das Funktionsprinzip von CMOS-Sensoren basiert auf einem ähnlichen Verfahren. Die lichtempfindlichen Elemente werden jedoch nicht über das Ladungsaustauschverfahren, sondern direkt über eine Vielzahl an Transistoren an den Pixeln ausgelesen. Abbildung 4.2 zeigt den Aufbau eines CMOS-Sensors.

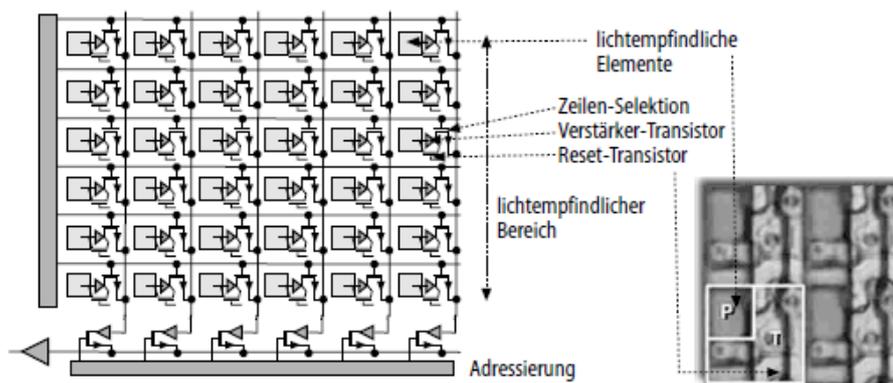


Abb. 6.8. CMOS-Funktionsprinzip und Ausschnitt aus der Sensorfläche

Abbildung 4.2.: Grundlegender Aufbau einer CMOS-Zellen [51, Abb. 6.8]

Die einfachste Form des Ausleseverfahrens ist mit einem Transistor pro Pixel, die in einer xy-Matrix angeordnet sind. Diese Konstruktion wird als Passive Pixel Sensor (PPS) bezeichnet. Moderne Sensoren verwenden meist eine Vielzahl an Transistoren, die unterschiedliche Aufgaben erfüllen. Das ermöglicht die Genauigkeit der Ladungserfassung und damit auch die Reduktion verschiedener Rauscharten, beispielsweise durch eine Correlated Double Sampling (CDS), also eine gezielte Doppelabtastung des Sensors um das Ausleserauschen zu minimieren. Zudem wird immer mehr mit integrierten A/D-Wandlern gearbeitet, die eine direkte Verarbeitung der Sensorströme ermöglichen. Daraus ergeben sich eine Reihe von Vorteilen von CMOS-Sensoren. Zum einen lassen sich durch die Verwendung von Speichertransistoren alle Pixel des Sensors zeitgleich auslesen (Global Shutter), wodurch temporale Artefakte (Rolling Shutter) in dem Ausleseverfahren verhindert werden können. Die Werte lassen sich zudem direkt über eine xy-Matrix zuweisen und müssen im Nachgang nicht über die Auslesereihenfolge rekonstruiert werden. Darüber hinaus lassen sich auch nur bestimmte Teilbereiche, auch als Region of Interest (RoI) bezeichnet, auslesen, was einen signifikanten Unterschied in der Handhabung der Datenmengen machen kann [51, S. 395-399].

Eine besondere Form der CMOS-Sensoren sind die Foveon Sensoren, die von der gleichnamigen Firma entwickelt und später von der Firma Sigma aufgekauft wurden. Photonen mit einem unterschiedlichen Energieniveau haben die Eigenschaft, in Halbleitermaterialien unterschiedlich tief einzudringen. Dadurch ist die Eindringtiefe abhängig von der Wellenlänge [51, S. 390-391]. Auf Grundlage dieses Effekts kann man innerhalb eines einzelnen Pixels die RGB-Grundfarben separieren, indem man den Sensor in der Tiefe staffelt und in drei Zonen einteilt [26]. Foveon-

Sensoren haben im Vergleich zu klassischen mosaikbasierten Sensoren einige entscheidende Vorteile. Durch die Erfassung des gesamten sichtbaren Spektrums auf jedem Pixel erhöht sich die Farbauflösung des Sensors im Vergleich zu mosaikbasierten Sensoren. Darüber ist das Risiko auf Moiré-Effekte bei Foveon-Sensoren deutlich verringert, da keine Farbinterpolation zwischen den Pixeln notwendig ist. Dadurch kann auch die Detailgenauigkeit und Schärfe des Sensors verbessert werden, ohne das Anti-Aliasing-Filter notwendig sind [51, S. 390-391] [26].

4.1.2. Abbildungsfehler elektrischer Kameraelemente

Elektrische Kameraelemente zeigen eine Reihe von Abbildungsfehlern, die im Prozess der Bildgewinnung und Umwandlung entstehen können. Im Folgenden soll auf die wichtigsten Abbildungsfehler eingegangen werden.

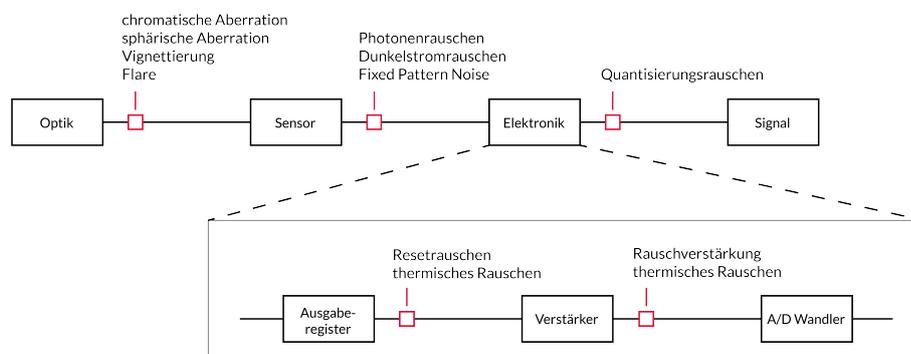


Abbildung 4.3.: Störeinflüsse beim Bildwandler und der Optik, frei nach [51, Abb. 6.31]

Abbildung 4.3 zeigt die Störeinflüsse bei Bildwandlern und Optiken. Auf die optischen Störeinflüsse wird noch einmal gesondert im Kapitel 4.2 eingegangen.

Das Rauschen einer Kamera ist eines der wichtigsten Qualitätsmerkmale bei Kamerasystemen. Fast jeder Umwandlungsschritt nimmt Einfluss auf das Rauschen einer Kamera. Es gibt verschiedene Rauscheffekte:

Das **Photonenrauschen** tritt auf, da Photonen einer statistischen Verteilung, der Poisson-Verteilung unterliegen. Die Photonen, die auf einen Sensor treffen, variieren zufällig in ihrer Position, was zu Schwankungen in der Helligkeitsverteilung über dem Pixelsensor führt. Je schwächer der Sensor beleuchtet wird, desto stärker fällt dieser Effekt relativ ins Gewicht [27] [51, S. 413-415].

Das **Dunkelstromrauschen** entsteht durch die Abhängigkeit der Elektronen im Halbleitermaterial von der Sensortemperatur. Auch ohne die Beleuchtung des Sensors werden durch die thermische Energie in einem Sensor Elektronen aus dem Halbleitermaterial freigesetzt. Je höher die Sensortemperatur ist, desto stärker tritt dieser

Effekt auf. Die Elektronen werden in den Pixeln gespeichert und sorgen so für ein thermisches Grundrauschen [27] [51, S. 413-415].

Das **Fixed Pattern Noise** ist eine Form des Rauschens, die systembedingt durch den Fertigungsprozess der einzelnen Pixel entsteht. Durch Schwankungen in der Fertigung der Pixel kommt es zu Unregelmäßigkeiten in der Empfindlichkeit und Sättigungskapazität der einzelnen Pixel. Diese Unregelmäßigkeiten führen zu konstanten Helligkeitsmustern über alle Pixel. Das Fixed Pattern Noise lässt sich durch nachträgliche Korrekturen dieser Unregelmäßigkeiten relativ einfach beheben [48] [51, S. 413-415].

Das **Ausleserauschen** tritt beim Auslesen der elektrischen Signale von den Pixeln auf. Durch die elektrischen Schaltungen und Bauteilen kommt es zur Rauschbildung. Ausleserauschen tritt bei verschiedenen Bauteilen wie den Transistoren, Signalverstärkern und Analog-Digital-Wandlern auf. Ein Verstärker, der das schwache Sensorsignal verstärkt, kann diese Verstärkung beispielsweise nicht rauschfrei tun. Jede Verstärkerstufe fügt dabei dem Rauschsignal zusätzliches Rauschen hinzu. Das spezifische Rauschen ist stark von den verwendeten Bauteilen abhängig. In der Regel tritt das Ausleserauschen jedoch als konstanter Rauschwert auf. Dadurch hat das Rauschen besonders dann großen Einfluss, wenn die Helligkeit der Szene gering ist.

Das **Resetrauschen** entsteht bei dem Zurücksetzen der Pixel. Um die Pixel zurückzusetzen, wird ein Reset-Transistor verwendet. Wenn der Transistor gepulst wird, wird er durch den angelegten Strom leitfähig. Bei diesem Vorgang entladen sich die Pixel. Der Transistor selbst agiert dabei wie ein niederohmiger Widerstand. Dies kann beim Entladevorgang zu thermischem Widerstandsrauschen führen. Das Resetrauschen tritt als weißes Rauschen über alle Frequenzen des Signals auf, kann jedoch durch die Verwendung von CDS vollständig entfernt werden [21] [50]. Bei dem CDS wird die Pixelspannung doppelt gemessen. Einmal beim Zurücksetzen des Pixels und einmal unmittelbar nach dem Reset. Die Differenz ergibt das bereinigte Signal [49].

Das **thermische Rauschen**, auch als Nyquist-Rauschen bezeichnet, ähnelt sehr stark dem Dunkelstromrauschen, entsteht jedoch auf Grund eines anderen Effekts. Während beim Dunkelstromrauschen durch thermische Prozesse im Halbleitermaterial Elektronen freigesetzt werden, entsteht das thermische Rauschen durch die zufällige Bewegung von Ladungsträgern in leitenden Materialien. Es ist ebenfalls temperaturabhängig und kann in allen elektronischen Bauteilen auftreten. Je höher die Temperatur dabei ist, desto stärker ist auch das thermische Rauschen. Es kann wie das Dunkelstromrauschen durch die Temperaturregelung des Sensors minimiert werden.

Das **Quantisierungsrauschen** tritt immer dann auf, wenn ein analoges Signal durch einen Analog-Digital-Wandler in ein digitales Signal umgewandelt wird. Da in digitalen Signalen immer nur eine begrenzte Anzahl an Bits verwendet wird, kommt es

bei der Umwandlung zu Signalverlusten. Die Differenz zwischen dem ursprünglichen analogen Signal und dem digitalisierten Signal ergibt das Quantisierungsrauschen. Das Quantisierungsrauschen kann durch die Verwendung von vielen Bits minimiert werden [51, S. 413-415].

Neben den hier aufgeführten Rauschformen gibt es noch eine Reihe weiterer Effekte, wie das Kanalrauschen bei CCD-Sensoren oder das kohärente Rauschen durch Störungen durch elektromagnetische Einflüsse, diese sind aber für die folgenden Betrachtungen vernachlässigbar.

Neben den Rauscheffekten können bei der Herstellung von Sensoren durch Unregelmäßigkeiten im Fertigungsprozess Abbildungsfehler entstehen, die zu einer Verschlechterung der Sensorgenauigkeit und Sensorkapazität führen. Dazu zählen unter anderem die Dotierung des Halbleitermaterials, die Pixelgröße und genaue Ausrichtung der Pixel, die Schichtdicke in den Photodioden sowie Defekte und Verunreinigungen [29] [18].

Von Dotierung spricht man bei Halbleitermaterialien bei dem Einbringen von Fremdatomen in ein Halbleiterkristall. Durch die Dotierung kann man die Leitfähigkeit von Halbleitermaterialien beeinflussen. Die Dotierung selbst kann nur mit einer gewissen Schwankung realisiert werden, was zu Unterschieden in der Leitfähigkeit von Halbleiterkristallen führen kann. Diese Unterschiede können Einfluss auf das Signal und die Dunkelströme haben, wodurch es zu Unterschieden vor allem in dem Rauschsignal kommen kann.

Durch die Variationen in der Pixelgröße und Schichtdicke in den Pixeln, kann es zu Unterschieden in der Sensorkapazität kommen. Auch hier unterliegen die Pixel im Fertigungsprozess Schwankungen. Je größer bzw. dicker die Pixel eines Sensors sind, desto mehr Photonen können aufgenommen werden, was die Sensorkapazität erhöht.

Verunreinigungen beeinflussen die Pixelempfindlichkeit des Sensors. Bei den verunreinigten Pixeln kommt es zu Abschattung auf den Pixeln, dadurch können sie teilweise oder vollständig von den Lichtphotonen abgeschirmt werden. Bei konstanter Beleuchtungsdichte fallen dadurch weniger Photonen auf die verunreinigten Pixel.

Neben Variationen und Verunreinigungen können Pixel oder Pixelbereiche auch gänzlich defekt sein. Hier unterscheidet man zwischen toten Pixeln und Hotpixeln. Tote Pixel sind Pixel, die permanent keine Daten liefern. Sie können entstehen, wenn die Pixel des Sensors entweder technisch defekt sind, oder sich elektrische Defekte im Ausleseprozess befinden. Tote Pixel erscheinen als dunkle Punkte im Bild. Von Hotpixeln spricht man, wenn Pixel dauerhaft zu hohe Signalwerte liefern. Grund hierfür kann eine Überempfindlichkeit des Sensors beispielsweise durch eine fehlerhafte Dotierung sein, oder wenn durch eine erhöhte Spannung die Pixelwerte zu stark verstärkt werden. Hotpixel treten als helle oder farbige Punkte im

Bild auf.

Auch nach der A-D-Wandlung kann es zu verschiedenen Abbildungsfehlern durch die Dateninterpretation kommen. Bei der Kameracharakterisierung wird das Bildsignal in vielen verschiedenen Schritten für die Wiedergabe optimiert. Abbildung 4.4 zeigt die unterschiedlichen Schritte in der Bildpipeline von Kamerasystemen.

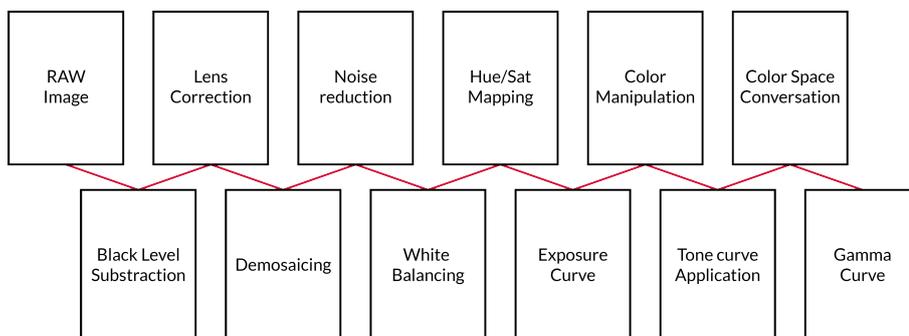


Abbildung 4.4.: Bildpipeline von Kamerasystemen, frei nach [39]

Unterschiedliche Kamerahersteller charakterisieren ihre Kameras auf unterschiedliche Weise. Manche Transformationsschritte dienen der technischen Transformation der Kameradaten in darstellbare Farb- und Luminanzräume, andere optimieren kameraspezifische Abbildungsfehler. Wieder andere verfolgen das Ziel, dem Kamerasystem einen herstellerspezifischen Look zu verpassen. Diese Transformationen führen zu Farbverschiebungen in den Bilddaten. Jede Kamera bildet Farben leicht unterschiedlich ab. Diese Color Shifts können ungewünscht durch Artefaktbildung wie beim Demosaicing entstehen oder durch Farbverschiebungen bewusst hervorgerufen werden.

Neben den genannten gibt es noch eine Vielzahl an weiteren Abbildungsfehlern, die jedoch für diese Arbeit eine untergeordnete Relevanz haben. Zu ihnen zählen unter anderem Aliasing Effekte wie der Moiré-Effekt, Rolling-Shutter, Blooming, Banding, Quantisierungsfehler und Kompressionsartefakte [51].

4.2. Eigenschaften optischer Kameraelemente

4.2.1. Optik

Kameraoptiken dienen in der Kameratechnik der Bündelung von Lichtstrahlen. Sie bestehen aus einem Linsensystem, welches die optische Projektion auf die Sensoroberfläche reguliert. Ziel ist es dabei, die Lichtverteilung in der Szene möglichst fehlerfrei abzubilden. Abbildung 4.5 zeigt den schematischen Aufbau von Kameraoptiken und dem Optischen Teiler in Broadcastkameras.

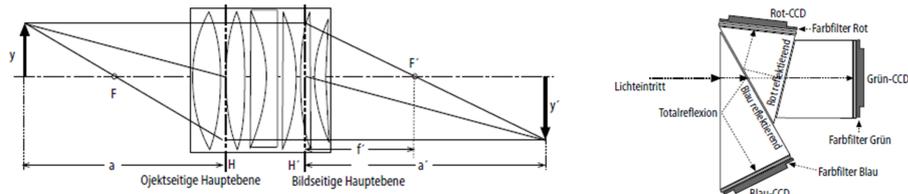


Abbildung 4.5.: Kameraoptiken von Broadcast- und Consumerkameras

In Kameraoptiken kommen verschiedene Linsentypen zum Einsatz, die die Abbildungsqualität auf verschiedene Weisen beeinflussen. Konkave und konvexe Linsen streuen und fokussieren das Licht. In Abbildung 4.5 sieht man wie sie in Kombination verwendet werden, um Lichtstrahlen auf einen bestimmten Bereich zu fokussieren. Mit Hilfe von asphärischen und achromatischen Linsen lässt sich der Bildausschnitt sphärisch und chromatisch korrigieren. Sie werden verwendet, um bestimmte Abbildungsfehler zu korrigieren. Die Vergütung der einzelnen Linsen spielt ebenfalls eine entscheidende Rolle bei der Abbildungsqualität. An schlecht vergüteten Linsen kann es durch Reflexionen an der Glasoberfläche zu Abbildungsfehlern kommen. Eine genauere Betrachtung der optischen Abbildungsfehler folgt im folgenden Kapitel [51, S. 440-444].

Durch die Kombination aus verschiedenen Linsen werden die Objektive für die Verwendung mit bestimmten Kameraobjektiven optimiert. Im Folgenden werden kurz die wichtigsten Kamerafunktionen erläutert:

Die **Brennweite** einer Kameralinse ist definiert über den Abstand der Brennebene zur Optik. Die Brennebene beschreibt die Tiefenebene, in dem sich die Lichtstrahlen im Brennpunkt kreuzen. Je kürzer die Brennweite ist, desto weitwinkliger ist der Bildwinkel [51, S. 449-452].

Die **Blende** reguliert mechanisch den Lichteinfall auf den Sensor. Sie besteht aus einzelnen Blendenlamellen, die geöffnet und geschlossen werden. Je größer die Blendenöffnung, desto mehr Licht kann auf den Sensor fallen. Durch Abschattung an den Rändern kann es bei einer großen Blendenöffnung jedoch zu Vignettierungseffekten kommen. Zudem sinkt die Schärfentiefe, was Objekte außerhalb des Fokuspunktes unschärfer wirken lässt. Bei stark geschlossenen Blenden trifft entsprechend weniger Licht auf den Sensor, jedoch steigt die Schärfentiefe des Sensors. An den Blendenlamellen kommt es durch die geringe Öffnung vermehrt zu Beugungseffekten, die die Schärfe beeinträchtigen können [51, S. 444-445].

Der bereits erwähnte **Schärfebereich** beschreibt den Teil des Bildes, der für das menschliche Auge scharf wirkt. Für den durchschnittlichen Menschen liegt dieser Bereich in einem Zerstreuungskreis von $1/30$ mm. Der Zerstreuungskreis wird definiert durch den Abstand der äußersten Lichtstrahlen zueinander. Der Punkt mit einem Zerstreuungskreis von $1/30$ mm wird als Nahpunkt bzw. Fernpunkt bezeichnet. Der Abstand dazwischen definiert den Schärfebereich der Optik. Abbildung 4.6 zeigt den Schärfebereich bei unterschiedlichen Blendenöffnungen [8, S.

38] [51, S. 445-447].

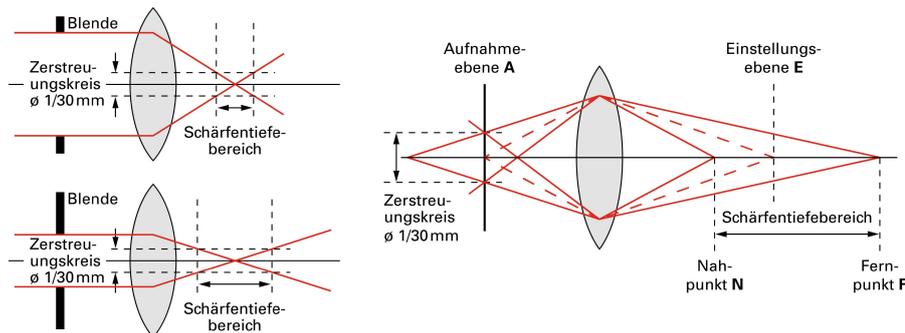


Abbildung 4.6.: Die Schärfentiefe von Kameraoptiken [8, S. 38]

4.2.2. Abbildungsfehler optischer Kameraelemente

Durch verschiedene physikalische Effekte wie der Dispersion, Brechung und Beugung, sowie ungewollter Reflexionen auf den Linsenoberflächen kommt es in der Kameraoptik zu Abbildungsfehlern. Eine genaue Betrachtung der Effekte findet sich in Kapitel 2. Im folgenden werden die für diese Arbeit relevantesten Abbildungsfehler erläutert.

Die **chromatische Aberration** entsteht durch die Dispersion von Lichtstrahlen unterschiedlicher Wellenlängen durch die Linsen. Sie äußert sich durch Farbverschiebungen im Bild, die vor allem als blaue oder rote Ränder an kontrastreichen Kanten zu sehen sind. Sie entstehen dadurch, dass die kurzwelligen Anteile des Lichts stärker gebrochen werden als die langwelligen Anteile. In der Optik nutzt man apochromatische Linsen oder verschiedene Kombinationen aus Linsen, um den Fehler zu minimieren. [30, S. 94-95] [51, S. 440-442].

Die **sphärische Aberration** ist ein Schärfefehler und tritt auf, wenn Lichtstrahlen an den Rändern von sphärischen Linsen stärker gebrochen werden, als in der Linsenmitte. Das führt dazu, dass Lichtstrahlen, die weiter vom Zentrum der Linse entfernt sind, nicht auf den Brennpunkt, sondern kurz davor oder kurz dahinter fokussiert werden. Dieser Effekt führt zu Unschärfe im Bild. Mit Hilfe von asphärischen Linsen lassen sich unterschiedliche Lichtstrahlen unterschiedlich brechen, wodurch das Problem der sphärischen Aberration minimiert werden kann [30, S. 94-95] [51, S. 440-442].

Vignettierung entsteht immer dann, wenn die Ränder des Sensors schwächer beleuchtet werden als die Sensormitte. Es gibt verschiedene Ursachen für die Vignettierung. Mechanische Bauteile wie der Objektivtubus oder einzelne Linsenelemente können die Ränder des Objektivs abschatten. Das passiert vor allem immer dann, wenn mit einer weiten Brennweite aufgezeichnet wird. Auch das optische Design

der Linsen kann Einfluss auf die Vignettierung nehmen. Treffen Lichtstrahlen beispielsweise an den Linsenrändern in einem steileren Winkel auf die Sensoren, führt das zu Energieverlusten [30, S. 94-95] [51, S. 440-442].

Der **Flare** bei Kameraoptiken entsteht durch ungewollte Lichtreflexionen auf den Linsen der Optik oder durch Streuungen im Inneren des Objektivs. Sie treten vor allem dann auf, wenn sehr helle Lichtquellen wie die Sonne direkt oder knapp außerhalb des Bildfelds der Kamera in das Objektiv scheinen. Durch die Reflektion zwischen zwei Linsen kommt es dabei zu Lichtflecken, die unkontrolliert auf die Sensoroberfläche leuchten. Auch Verschmutzungen oder Beschädigungen an den Linsen können dafür sorgen, dass Licht unkontrolliert gestreut wird. Zum Vermeiden von Flare Effekten spielt die Vergütung eine wichtige Rolle. Durch Beschichtungen der Linsen kann die Reflektanz verringert und dadurch die Entstehung von Flare Effekten vermieden werden. Außerdem können Gegenlichtblenden direktes Licht abschirmen, das knapp außerhalb des Bildfelds liegt.

Neben den genannten Abbildungsfehlern gibt es auch noch viele weitere Fehler, die vor allem mit der optischen Schärfe und Wölbung der Abbildung zu tun haben, die jedoch für die weiteren Ausführungen nicht betrachtet werden. Dazu zählen unter anderem das Koma bei Kameras, der Astigmatismus, Verzeichnungen, verschiedene Beugungseffekte durch Linsen und die Kamerablende, Bildfeldwölbungen sowie Unschärfefeffekte.

4.3. Qualitätsmerkmale von Kamerasystemen

Aus den Abbildungsfehlern der elektrischen und optischen Kameraelemente lassen sich einige Qualitätsmerkmale ableiten. Es gibt eine ganze Reihe von Qualitätskriterien, die definiert werden können, weit mehr, als im Rahmen dieser Arbeit berücksichtigt werden können. In der folgenden Auflistung sind die wichtigsten Kriterien aufgeführt [34] [14].

Im Bereich der Bildqualität können Kamerasysteme durch die Schärfe, das Rauschen, den Tonwertumfang, den Kontrast, die Farbgenauigkeit, den Signal-Rauschabstand, den Dynamikumfang, die Texturdetails, die Auflösung und die Texturverluste definiert werden. Im Bereich der optischen Eigenschaften der Kameraobjektive können Systeme nach dem Grad der Verzeichnung, der Vignettierung, den Inhomogenitäten, den chromatischen und sphärischen Aberrationen, den Glanzlichtern und den Aliasing-Effekten eingeteilt werden. Die Belichtungsgenauigkeit, ISO-Empfindlichkeit und Leistung bei schwachem Licht kann unter den Punkt Belichtung zusammengefasst werden. Bei den Artefakten lassen sich Blemishes, Datenkompressionsverluste und Software-Artefakte zusammenfassen. Zudem lassen sich Kameras durch ihre Bildstabilisierung, Weißabgleich-Genauigkeit, Autofokus-Geschwindigkeit und Genauigkeit, Sensorgröße, Pixeldichte und Farbwiedergabe klassifizieren.

In dieser Arbeit wird auf die luminanz- und spektralspezifischen Qualitätsmerkmale eingegangen und die Kamerasysteme auf diese überprüft. Sie sind im Folgenden definiert.

Um luminanzspezifisch Aussagen treffen zu können, müssen die **Gammakurven** von Kamerasystemen linear sein. Nur so kann gewährleistet werden, dass die Belichtungssituationen auf alle Kamerasysteme den gleichen Einfluss haben. Daher ist die Linearität ein elementares Qualitätskriterium für die Messung von Kamerasystemen.

Auch die **Vignettierung**, die Messergebnisse von Farb- und Lumachartmessungen beeinflussen kann, ist für die Aussage von Helligkeits- und Farbgenauigkeiten essenziell. Über die Helligkeitsunterschiede zwischen den verschiedenen Pixeln des Sensors und vor allem zwischen Pixeln in der Sensormitte zu Randpixeln kann das Maß für die Vignette gezeigt werden.

Wie bei den Abbildungsfehlern elektrischer Kameraelemente gezeigt wurde, gibt es eine Vielzahl an **Rauscheffekten**, die Einfluss auf die Bildqualität eines Kamerasystems nehmen. Das Rauschverhalten im Bezug auf das Gesamtrauschen kann kameraübergreifend verglichen werden. Darüber hinaus lässt sich das Rauschverhalten bei sich unterscheidenden Vorverstärkungen und damit simulierten Lichtsituationen vergleichen.

Resultierend aus dem Rauschmuster von Kameras lässt sich der **Signal-Noise-Ratio (SNR)** ermitteln. Der SNR ist ein Maß für das Verhältnis von dem Bildsignal zum Rauschen.

Kamerasysteme zeigen verschiedene Inhomogenitäten. Sie lassen sich sowohl für die Bild- als auch Rauschsignale spezifizieren und werden in der **DSNU** und **PRNU** ermittelt. Die DSNU zeigt die Inhomogenitäten in Bezug auf das Dunkelsignal des Sensors, also wenn kein Licht auf den Sensor fällt. Auch wenn keine Photonen auf den Sensor fallen, kommt es durch verschiedene thermische und quantentechnische Effekte zu einem Dunkelstrom. Das Verhalten des Sensors im Bezug auf den Dunkelstrom wird in der DSNU zusammengefasst. Die PRNU beschreibt die Inhomogenitäten zwischen den einzelnen Pixeln bei gleicher Beleuchtung. Durch die Unterschiede in der Empfindlichkeit und Sensorkapazität kommt es zu Abweichungen. Diese werden in der PRNU abgebildet [18] [48].

Durch die Unterschiede in den Empfindlichkeiten einzelner Sensel können lokale aber auch globale Farbverschiebungen entstehen. Die Darstellungsgenauigkeit von Farben ist demnach ein weiteres Qualitätskriterium für Kamerasysteme. Sie wird auch als **Farbtreue** bezeichnet. Neben den sensorbezogenen Verschiebungen kann es auch zu Verschiebungen durch die Bildpipeline kommen (siehe Abbildung 4.4). Die Farbverschiebungen können durch den Hersteller gewollt sein und als Look auf die Kameras angewendet werden. Hier lassen sich demnach nur die Kameradaten mit einer simulierten Referenzkamera vergleichen.

5. Messung

In diesem Kapitel wird der Messaufbau für die Kameramessungen detailliert beschrieben und die Vorgehensweise der Datenerhebung erklärt. Insgesamt wurden acht verschiedene Kamerasysteme (drei Broadcastkameras, drei Consumerkameras, eine Chipkamera und eine Drohne) mit dem Messaufbau gemessen.

5.1. Methodik

Um die in den Grundlagenkapiteln hergeleiteten Forschungsfragen zu beantworten, wurde in der vorliegenden Abschlussarbeit eine Kombination aus quantitativer Studie und Fallstudie durchgeführt. Aus den Grundlagen wurden Anforderungen an Kamerasysteme abgeleitet, die sowohl in Bezug zum HVS als auch im Bezug zu HDR-WCG-Produktionen gelten.

Die Anforderungen an das menschliche Wahrnehmungssystem lauten:

- Kamerasysteme müssen szenenlinear einen Blendenumfang von 12,3 Blendenstufen erreichen, um die Wahrnehmungsgrenze des steady state des HVS zu erreichen.
- Kamerasysteme müssen einen Signal-Rauschabstand von mindestens 42 dB vorweisen, um für das visuelle System die ungewichtete Wahrnehmungsgrenze des Rauschens nicht zu überschreiten.
- Kamerasysteme müssen eine Farbdifferenz von kleiner $1 \Delta E$ im $L^*a^*b^*$ Farbraum zu den Referenzmessdaten vorweisen, um keine visuellen Unterschiede zuzulassen.
- Kamerasysteme müssen untereinander eine Farbdifferenz von kleiner $1 \Delta E$ im $L^*a^*b^*$ Farbraum vorweisen, um keine visuellen Unterschiede zwischen den Kamerasystemen zuzulassen.

Die Anforderungen an HDR-WCG-Produktionen lauten:

- Kamerasysteme müssen szenenlinear einen Dynamikumfang von 6,74 Blendenstufen abbilden können, um für einen SDR-Workflow geeignet zu sein.

- Kamerasysteme müssen szenenlinear einen Dynamikumfang von 14,67 Blendenstufen abbilden können, um für einen HLG-Workflow geeignet zu sein.
- Kamerasysteme müssen szenenlinear einen Dynamikumfang von 11,02 Blendenstufen abbilden können, um für einen PQ-Workflow mit einer Spitzenleuchtdichte von 1000 cd/m² geeignet zu sein.
- Kamerasysteme müssen szenenlinear einen Dynamikumfang von 12,82 Blendenstufen abbilden können, um für einen PQ-Workflow mit einer Spitzenleuchtdichte von 4000 cd/m² geeignet zu sein.
- Kamerasysteme müssen szenenlinear einen Dynamikumfang von 14,02 Blendenstufen abbilden können, um für einen PQ-Workflow mit einer Spitzenleuchtdichte von 10000 cd/m² geeignet zu sein.

Die Datenerhebung der Messdaten für die verwendeten Kamerasysteme erfolgt in vier Schritten. Im ersten Schritt werden die verwendeten Messinstrumente mit Hilfe der Referenzdaten des Herstellers verglichen. Im zweiten Schritt werden numerische Messdaten der Kamerasysteme erhoben. Im dritten Schritt werden die Messdaten ausgewertet. Im vierten Schritt werden die ausgewerteten numerischen Daten verwendet, um die Anforderungen an die Kamerasysteme für die jeweiligen Kameras zu bestätigen oder zu widerlegen. Eine genauere Beschreibung des Messaufbaus und der Messdurchführung folgt in diesem Kapitel. Die Auswertung und Interpretation der Daten folgt in Kapitel 6.

5.2. Messaufbau

Die Messungen wurden in dem Zeitraum vom 31.07.2024 bis zum 09.08.2024 durchgeführt. Sie fanden in einem Dunkelraum der Lang AG in Lindlar statt. Der Dunkelraum ist für präzise Kameramessungen ausgelegt. Entsprechend ist der Raum vollständig schwarz gestrichen und temperaturgeregelt. Er wurde über den gesamten Messzeitraum auf einer Raumtemperatur von 22° Celsius gehalten. Abbildung 5.1 zeigt Fotos des Messaufbaus innerhalb des Dunkelraums.

Wie auf den Bildern zu sehen ist, setzt sich der Messaufbau aus verschiedenen Komponenten zusammen. Insgesamt wurden sieben Messreihen aufgezeichnet (vgl. 5.4). Als Messinstrumente kamen der **Esser Test Charts Illuminator LE6-100** in Kombination mit den Messcharts **TE188** und **TE223** von Image Engineering zum Einsatz. Für die Verifikation der Messinstrumente wurden mit dem **SpektraScan PR-670** die Luminanz und die spektralen Eigenschaften der Lichtquelle und der Testcharts verifiziert. Eine genauere Betrachtung der verwendeten Messgeräte findet sich im Abschnitt 5.2.1.



Abbildung 5.1.: Der Messaufbau im Dunkelraum der Lang AG

Insgesamt wurden acht Kamerasysteme vermessen. Dazu zählen die folgenden Kamera-Objektiv Kombinationen:

Kamera	Objektiv	Kameratyp
Grass Valley LDX 135	Fujinon UA18x7.6BERD-S10B	Broadcast
Panasonic UC4000	Fujinon UA18x7.6BERD-S10B	Broadcast
Ikegami UHK-X700	Fujinon UA18x7.6BERD-S10B	Broadcast
Sony A7IV	Sony FE 2.8/24-70mm GM II	Consumer
Sony A7III	Tamron 28-75mm F/2.8 Di III RXD	Consumer
Panasonic Lumix GH5s	Sigma 18-35mm 1:1,8 DC	Consumer
GoPro HERO 10	-	Chip
DJI Phantom 4 Pro	-	Drohne

Tabelle 5.1.: Gemessene Kamera-Objektiv-Kombinationen

Bei den ersten drei Kameras handelt es sich um klassische Broadcastkameras. Die Grass Valley LDX 135 wurde über die Camera Control Unit (CCU) **Grass Valley XCU XPress UXF XF Fiber** angesteuert und mit Strom versorgt. Mit dem Remote Control Panel (RCP) **Grass Valley CGP500** wurden die Einstellungen an der Kamera vorgenommen. Die Panasonic UC4000 wurde über die **Panasonic AK UCU600** CCU mit dem **Panasonic AK-HRP1010** angesteuert. Die Ikegami UHK-X700 wurde über die Ikegami BSX-100 CCU und die Ikegami OCP-300 angesteuert. Bei der RCP der Grass Valley kam die Besonderheit hinzu, dass sie über den TP-Link TL-SG1005P mittels Power over Ethernet (PoE) mit Strom versorgt wurde. Alle Videodateien der Broadcastkameras wurden mit Hilfe eines **Atomos Ninja V** mit dem **SDI Modul** in ProRes-HQ aufgezeichnet. Die Messdaten der restlichen Kameras wurden per integrierter SD-Karte direkt in der Kamera aufgezeichnet.

Abbildung 5.2 zeigt das Blockschaltbild des Messaufbaus.

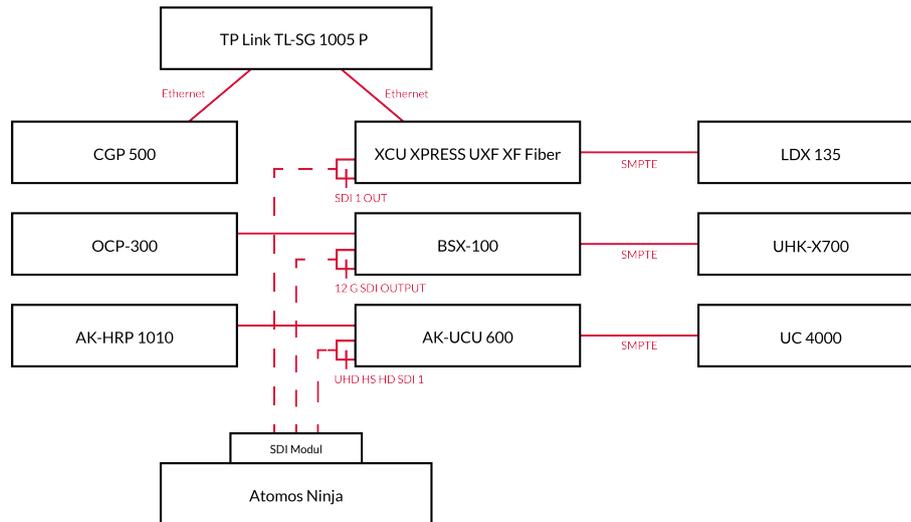


Abbildung 5.2.: Blockschaltbild des Messaufbaus

5.2.1. Messgeräte

Für den Messaufbau kamen verschiedene Messgeräte zum Einsatz. Im Folgenden werden die Funktionen und die Eigenschaften aller Messgeräte genauer erläutert.

Bei dem **Esser Test Charts Illuminator LE6-100** handelt es sich um eine spherische Lichtquelle (Ulbrichtkugel), die für die gleichmäßige Bestrahlung von transparenten Messcharts verwendet wird. Die Kugel ist mit einer 12V Halogenlampe ausgestattet (Osram HLX 64625), die laut Herstellerangaben über das gesamte Farbspektrum eine gleichmäßige Farbtemperatur von 3200 Kelvin (+-50 K) und eine gleichmäßige Abweichung der Beleuchtungsdichte des Messcharts von maximal 4 Prozent erreicht. Eine Überprüfung der Herstellerangaben wird in Abschnitt 5.4 durchgeführt. Das emittierte Licht wird über ein LCD Display an der Vorderseite eingestellt. Die Helligkeit des LE6 kann dabei zwischen 10 und 6000 Lux über einen mechanischen Verschluss eingestellt werden. Gemessen wird die Helligkeit über eine in die Kugel integrierte Photodiode. Daher ist die Helligkeit auch abhängig davon, wie viel Licht durch Reflektionen in die Kugel zurückgeworfen wird. Dies ist bei der Positionierung reflektierender Gegenstände vor dem LE6, wie beispielsweise der Messcharts zu beachten [15].

Bei dem Messchart **TE223** handelt es sich um ein logarithmisches Graustufentestchart, welches aus 13 Luminanzpatches mit unterschiedlich starker optischer Dichte besteht. Das Testchart hat ein Kontrastverhältnis von 200:1. Die Helligkeitsstufen des Herstellers werden in Tabelle A.1 im Anhang angegeben [17]:

Das **TE188 Testchart** ist auch als Color Checker bekannt und besteht aus 24 Farb- und Luminanzpatches mit vorgegebenen Farbpositionen. Eine Übersicht der Farbpositionen findet sich ebenfalls im Anhang in Tabelle A.3. Das Chart besteht aus zwölf Farbpatches, die verschiedene produktionskritische Situationen wie dunkle

und helle Hauttöne, den blauen Himmel, etc. abbilden. Dazu kommen sechs Farbpatches, die Primär- und Sekundärfarben abbilden, sowie sechs Graustufenpatches mit unterschiedlichen optischen Dichten.

Der **SpectraScan PR-670** der Firma JadaK ist ein portables Spektroradiometer. Es wird für die Spektralmessung vorwiegend von Lichtquellen und für die präzise Kalibrierung von Wiedergabesystemen verwendet. Der SpectraScan verfügt über 256 Detektorelemente und kann somit eine spektrale Auflösung von 1,56nm über den gesamten wahrnehmbaren Wellenlängenbereich von 380-780nm erfassen [38]. Der SpectraScan kann einen Dynamikumfang von 0.2 Footlambert (fL) bis 2500000 fL messen [37]. Das entspricht circa 0,7 bis 8565648 cd/m², was umgerechnet 23.6 Blendenstufen entspricht. Das Gerät wurde vor dem Messaufbau in regelmäßigen Abständen kalibriert. Für die Messungen diente das Gerät als Referenz für die Luminanz- und Spektralmessungen.

5.2.2. Kamerasysteme

Wie in Tabelle 5.1 beschrieben, wurden verschiedene Kamerasysteme miteinander verglichen. Diese teilen sich in zwei Messgruppen ein.

Die drei Broadcastkameras (Grass Valley LDX 135, Panasonic UC4000 und Ikegami UHK-X700) dienen als Referenzkameras und sollen Unterschiede verschiedener Kamerahersteller aufzeigen, die im Broadcastsegment aktiv sind. Die Kamerasysteme stellen keine repräsentative Stichprobe des Kameramarktes dar, sind aber sehr wohl drei viel genutzte Kamerasysteme und können den Messansatz damit gut erläutern. Hier gilt es zu erwähnen, dass andere Kamerasysteme auf Grund der Verfügbarkeit leider aus dem Messaufbau herausgefallen sind. Beispielsweise mit der LDX 86n von Grass Valley oder der HDC-4800C von Sony sollte der Datensatz noch erweitert werden.

Broadcastkameras haben die Eigenschaft, dass ihr Ausgangssignal durch eine Reihe von Einflussfaktoren vorverarbeitet wird. Ein Teil dieser Einflussfaktoren ist durch den Anwender konfigurierbar, ein anderer Teil jedoch auf Grund der Charakterisierung der Kameras durch den Hersteller nicht (vgl. 4). Ziel dieser Messreihe war es, die konfigurierbaren Einflussfaktoren vollständig auszuschalten und die Kameras anhand ihrer vom Hersteller definierten Eigenschaften zu vergleichen. Eine genaue Übersicht über die Kameraeinstellungen finden sich in den jeweiligen Versuchsdurchführungen.

Die zweite Messgruppe umfasst die der Consumerkameras. Anders als bei den Broadcastkameras lässt sich hier auf die Rohdaten der Kamerasensoren zugreifen. Dadurch können Qualitätskriterien der Kameras aktiv miteinander verglichen werden. In dieser Messgruppe werden die Kameras direkt auf verschiedene Qualitätskriterien von Kamerasystemen verglichen. Eine genauere Beschreibung findet sich

im Abschnitt 5.3.

Die GoPro HERO 10 und die DJI Phantom 4 Pro dienen als Vergleichskameras, um Unterschiede zwischen Broadcast-, Consumer-, Drohnen- und Chipkameras sichtbar zu machen.

5.3. Messansatz

Insgesamt wurden sieben Messreihen mit den Kameras durchgeführt. Die ersten vier Messungen dienen der Verifizierung des Messequipments. Dabei werden ausgehend von dem kalibrierten Spektrometer SpectraScan PR-670 als Referenzmessgerät alle verwendeten Messgeräte überprüft.

In der ersten Messung wird die Linearität des emittierten Lichts der Ulbrichtkugel überprüft, indem die gemessene Lichtstärke der integrierten Photodiode mit der gemessenen Beleuchtungsichte des Spektrometers verglichen wird. Dazu wurden sowohl eine lineare, als auch eine logarithmische Messreihe aufgezeichnet, um die Ulbrichtkugel für den linearen und logarithmischen Verlauf zu verifizieren und die Beleuchtungsichte für die entsprechenden Messwerte zu erheben.

In der zweiten Messung wurde die Farbkonstanz der Farbtemperatur der Lichtquelle verifiziert. Dies wurde ebenfalls für eine lineare und eine logarithmische Messreihe durchgeführt.

In der dritten Messreihe wurde für verschiedene Luminanzstufen die optische Dichte der einzelnen Lumapatches des TE223 mit der Referenz verglichen. Die originalen Referenzdaten finden sich in dem Usermanual des TE223. Vergleiche dazu die Referenzdaten in Tabelle A.1 und das Manual [17].

Die vierte Messreihe vergleicht die gemessenen Farbpositionen der Farbpatches des TE188 mit der Referenz aus dem Usermanual des TE188. Vergleiche dazu die Referenzdaten in der Tabelle A.3 und das Manual [16].

Die folgenden drei Messreihen dienen der Kameravermessung. Sie sind umfangreiche Messreihen, die sich auf die Luminanzeigenschaften der Kameras beziehen und die Farbgenauigkeit der Kameras überprüfen.

Dazu wurde in der fünften Messreihe die Gammakurve der Kameras vermessen. Dadurch kann die Linearität bei den Consumerkameras verifiziert und die Gammakurve der Broadcastkameras ermittelt werden.

Die sechste Messreihe misst den SNR bei verschiedenen Vorverstärkungen der Kamerasensoren. Dadurch können Kameraeigenschaften wie die Vignet-

tionierung, die DSNU, die PRNU, der Dynamikumfang und die Exposure Accuracy ermittelt werden.

In der siebten Messreihe wird die Farbgenauigkeit der Consumerkameras überprüft und der Kameralook der charakterisierten Broadcastkameras mit der Referenz im BT.2020 Farbraum eines HLG Signals verglichen.

Eine detaillierte Beschreibung der Rohdatenerhebung und Messdurchführung zu allen Messungen folgt im kommenden Abschnitt 5.4.

5.4. Messdurchführung

5.4.1. Verifikation der Messinstrumente

Messreihe 1: Linearität der Ulbrichtkugel

In der ersten Messung der Messreihe wurde die Linearität der Ulbrichtkugel durch die Messung der Beleuchtungsichte mit dem Spektrometer verifiziert. Dazu wurde das Spektrometer auf einem Stativ unmittelbar vor der offenen Ulbrichtkugel und entlang des Normalenvektors (90 Grad zur Oberfläche) zum Ausgang der Ulbrichtkugel positioniert. Abbildung 5.3 zeigt schematisch den Versuchsaufbau.

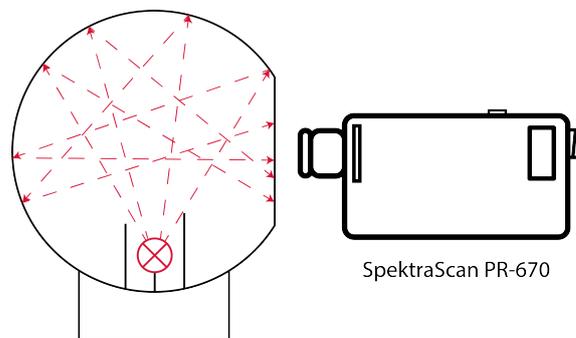


Abbildung 5.3.: Blockschaltbild der ersten und zweiten Messreihe

Über den gesamten Dynamikumfang wurden vier Messungen durchgeführt und die Beleuchtungsichten der Ulbrichtkugel mit dem Spektrometer in 200 Lux Schritten gemessen. Die Messungen waren zeitlich voneinander getrennt und wurden an unterschiedlichen Tagen durchgeführt. Anschließend wurden vier Messungen in logarithmischen Abständen durchgeführt. Der Versuchsaufbau bleibt unverändert. Auch hier sind die Messreihen zeitlich voneinander getrennt und wurden an unterschiedlichen Tagen durchgeführt. Im Anhang findet sich in Kapitel B.1 der vollständige Datensatz der gemessenen Beleuchtungsichten mit dem Spektrometer.

Messreihe 2: Farbkonstanz der Farbtemperatur des Leuchtmittels

In der zweiten Messreihe wurde die Farbkonstanz des Leuchtmittels überprüft. Dazu wurde mit dem Spektrometer die Farbtemperatur über verschiedene Lichtstärken der Ulbrichtkugel gemessen. Der Messaufbau gleicht dem Messaufbau in der ersten Messreihe und ist in Abbildung 5.3 dargestellt. Auch bei dieser Messreihe wurden zunächst vier Messungen mit linearen Helligkeitsabständen mit einer Lichtstärke von 200 Lux durchgeführt. Anschließend wurden vier Messungen in logarithmischen Helligkeitsabständen durchgeführt. Die Rohdaten der beiden Messungen finden sich in Kapitel B.2 im Anhang.

Messreihe 3: Messung der optischen Dichte des TE223

In der dritten Messreihe wurde die optische Dichte der einzelnen Graupatches des TE223 bei unterschiedlichen Helligkeitsstufen der Ulbrichtkugel vermessen. Dazu wurde das Chart in der Vorrichtung der Ulbrichtkugel und das Spektrometer direkt auf den Graupatches positioniert. Die nicht vermessenen Patches wurden durch schwarzen Molton abgedeckt um das Streulicht zu minimieren. Abbildung 5.4 zeigt den schematischen Aufbau der dritten Messreihe.

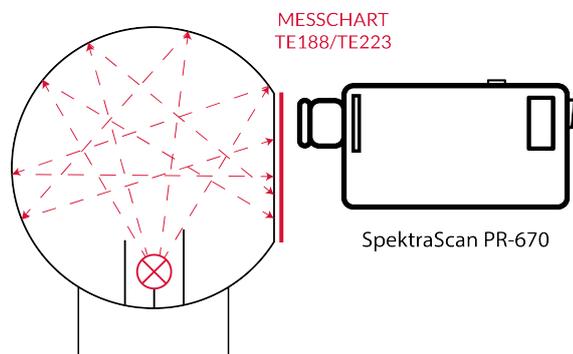


Abbildung 5.4.: Blockschaltbild der dritten Messreihe

Auch hier wurde das Spektrometer in Richtung des Normalenvektors des TE223 auf einem Stativ positioniert. Das Chart wurde für jeden Patch bei neun verschiedenen Helligkeitsstufen durchgemessen. Die Helligkeitsstufen sinken logarithmisch ab. Tabelle B.17 im Anhang zeigt die Rohdaten der Messung.

Messreihe 4: Messung der Farbpositionen der Farbpatches des TE188

In der vierten und letzten Messreihe zur Verifikation des Messmaterials wurden die Farbpositionen der Farbpatches des TE188 vermessen. Dazu wurde das Messchart in der Vorrichtung der Ulbrichtkugel positioniert. Der restliche Messaufbau

entspricht dem Aufbau aus Messreihe 3 und ist in Abbildung 5.4 dargestellt. Die Rohdaten der Messung finden sich in Tabelle B.18.

5.4.2. Luminanzmessungen der Kameras

Messreihe 5: Messung der Gammakurve der Kamerasensoren

Um eine Aussage über die Qualität von Kamerasensoren treffen zu können, muss sichergestellt sein, dass die Gammakurven der Kamerasysteme untereinander vergleichbar sind. Für die Kamerasysteme, die eine Rohdatenerhebung ermöglichen, lassen sich die Sensordaten direkt auswerten, da die Daten in der szenenlinearen Domäne vorliegen, wodurch ein linearer Zusammenhang zwischen der Beleuchtungsstärke der Lichtquelle und der Signalstärke der Sensel vorliegt. Bei den vorverarbeiteten Sensordaten wird durch die Gammakurve und die Funktionen der Kamera das Signal beeinflusst. Um sicherzustellen, dass die Kameras vergleichbar sind, wurde bei der fünften Messreihe die Gammakurve der Kamerasysteme ermittelt.

Um die Gammakurve zu ermitteln, wurden die Kameras auf eine feste Blende von F8.0 eingestellt, um den Einfluss durch Brechungs- und Beugungseffekte zu minimieren (vergleiche 2). Um den Einfluss durch die Sensorvorverstärkung zu minimieren, wurden die Consumerkameras mit einem ISO von 100 verwendet und die Broadcastkameras wurden auf 0dB Vorverstärkung eingestellt. Der Weißabgleich wurde, wie zuvor verifiziert, auf einen Weißpunkt von 3100 Kelvin eingestellt. Der Shutter speed wurde auf 1/60 Sekunden eingestellt. Bei einer Vollaussteuerung des Leuchtmittels der Ulbrichtkugel führen diese Einstellungen bei allen Kameras zum gewünschten Ausbrennen des Videosignals.

Zur Ermittlung der Gammakurve wurden für alle Helligkeitsstufen Messwerte in logarithmischen Abständen über den gesamten Dynamikumfang der Lichtquelle erhoben. Da die Lichtquelle in den dunklen Bereichen den Dynamikumfang der Kameras nicht voll ausreizen kann, wurden die Messungen mit dem TE223 durchgeführt und die Luminanzwerte für jede Helligkeitsstufe der Ulbrichtkugel mit dem Spektrometer gemessen. Eine Übersicht über die Messwerte finden sich im Anhang in Tabelle B.20. Es gilt zu beachten, dass das TE223 für Messungen der Gammakurve bei einem Systemgamma von 2,2 ausgelegt ist und es deshalb bei diesem Messverfahren zu ungleichmäßigen Abständen zwischen den Signalwerten des Sensors kommen kann. Gleichmäßige Helligkeitsabstufungen über den gesamten Abbildungsbereich der Kamera wären ideal, um eine Unterabtastung bestimmter Helligkeitsbereiche zu verhindern. Durch die Belichtungsreihe wird dieser Umstand bestmöglich kompensiert. Eine genauere Betrachtung findet sich in der Auswertung der Messdaten in Kapitel 6.

Die Broadcastkameras wurden alle auf eine Brennweite von 30 eingestellt, die Con-

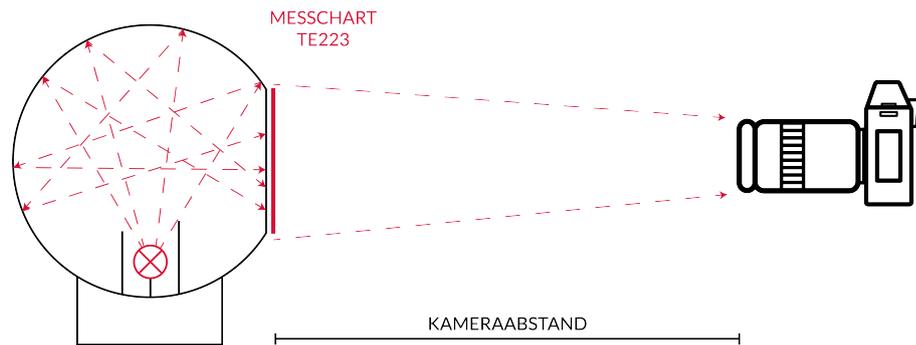


Abbildung 5.5.: Blockschaltbild der fünften Messreihe

sumerkamas wurden auf eine Brennweite von 70 eingestellt. Für die Lumix GH5s gilt zu beachten, dass durch den Cropfaktor des MFT-Sensors die Brennweite 35 einer Brennweite von 70 der Full-Frame Sensoren der Sony A7IV und Sony A7III entspricht. Die Broadcastkamas wurden in einem Abstand von ca. 100cm von dem Messchart entfernt positioniert. Die Consumerkamas wurden in einem Abstand von 70cm von dem Messchart entfernt positioniert. Da die GoPro und Phantom 4 Pro nur durch digitalen Zoom in der Brennweite beeinflusst werden kann, wurden beide Kamas so positioniert, dass das Testchart annähernd framefüllend positioniert worden ist, was einem Abstand von 10cm bei der GoPro und 15cm bei der Phantom 4 Pro entspricht. Abbildung 5.5 zeigt den schematischen Aufbau der Messreihe.

Um sicherzustellen, dass die Broadcastkamas möglichst geringen Einfluss auf die Gammakurve nehmen, wurden alle Kamas zunächst auf die Standardeinstellungen zurückgesetzt. Eine genaue Übersicht der Einstellungen beim Factory Reset findet sich in den Manuals [10] [62] [33]. Anschließend wurden alle Kamas auf eine Auflösung von Full HD bei 50 Frames pro Sekunde eingestellt. Die Gammakurve entspricht dem HDR-Standard HLG und als Referenzfarbraum dient BT.2020.

Für alle Broadcastkamas wurden anschließend die zusätzlichen Flare, Matrix, Gamma, Knee, Black sowie White Clip Einstellungen ausgeschaltet.

Anschließend wurde wie beschrieben bei neun verschiedenen Helligkeitsstufen des Leuchtmittels der Ulbrichtkugel eine Messung der Beleuchtungsichte der Lumapatches mit dem Spektrometer durchgeführt und bei allen Helligkeitsstufen eine Aufnahme des TE223 gemacht, mit jeder Kamera in einem individuellen Aufbau. Vereinzelt wurde bei Vollaussteuerung des Leuchtmittels der Ulbrichtkugel die Belichtungszeit des Sensors erhöht, um die ganz hellen Signalbereiche abzudecken.

Messreihe 6: Messung der Sensoreigenschaften der Kameras

Die sechste Messreihe zu den Sensoreigenschaften der Kamerasysteme wurde in zwei Messungen durchgeführt. Die Voreinstellungen der verschiedenen Kamerasysteme bleiben identisch und werden entsprechend der Messanforderungen für die einzelnen Messungen angepasst.

Bei der ersten Messung wurden die Signal-Rausch-Abstände der Kameras unter verschiedenen Vorverstärkungen des Sensors ermittelt. Die Belichtungszeiten und Blendeneinstellungen der Kameras bleiben dabei konstant. Die Helligkeitsunterschiede, die durch die Anpassung der Vorverstärkung entstehen, wurden durch die Belichtung der Ulbrichtkugel kompensiert. Dadurch werden die Signalwerte der Kameras konstant gehalten. Das erlaubt eine Aussage über den Signal-Rausch-Abstand bei unterschiedlichen Belichtungssituationen. Die Kameras wurden bei dieser Messung knapp vor dem Auslass der Ulbrichtkugel positioniert und der Fokus der Kamera auf unendlich gestellt. Dadurch minimiert man den Einfluss durch Staubpartikel auf der Linse oder auf der Innenbeschichtung der Ulbrichtkugel. Der schematische Aufbau dieser Messung findet sich in Abbildung 5.6, und die Rohdaten über die Beleuchtungsstärke und Isoeinstellungen der Kameras finden sich in Kapitel B.6 im Anhang.

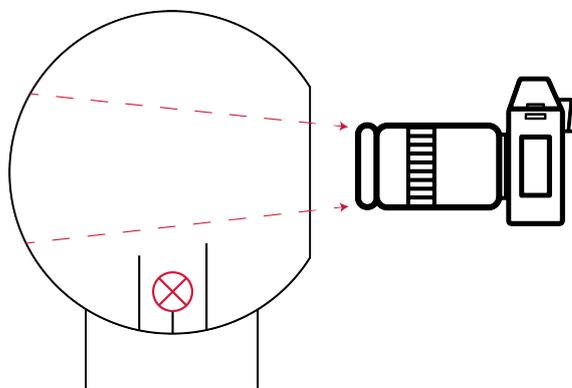


Abbildung 5.6.: Blockschaltbild der ersten Messung aus der sechsten Messreihe

Bei der zweiten Messung wurden die Kameraeinstellungen ebenfalls beibehalten und die Kameras auf das TE223 ausgerichtet und fokussiert. Der Abstand zum Messchart entspricht den Abständen in Messung 5.4.2. Der Fokuspunkt wurde dieses mal exakt auf das Messchart gerichtet. Auch bei dieser Messung wurden verschiedene Vorverstärkungen durchgemessen und die Helligkeit des Leuchtmittels der Ulbrichtkugel in logarithmischen Abständen angepasst. Die Rohdaten finden sich im Anhang B.6, und der schematische Messaufbau entspricht dem Messaufbau in Abbildung 5.6.

5.4.3. Spektralmessung der Kameras

Messreihe 7: Messung der Farbgenauigkeit der Kameras

In der letzten Messreihe wurden die Kameras auf ihre Farbgenauigkeit im Referenzfarbraum BT.2020 überprüft. Dazu wurden die Kameras wie bei Messung 5.4.2 auf das Farbchart TE188 fokussiert. Der Messabstand blieb identisch. Die Kameras wurden wieder bei einer Blende von F8.0 mit 1/60 Sekunde Belichtungszeit und einer Vorverstärkung bei ISO 100 bzw. 0dB verwendet. Die Belichtungsstärke des Leuchtmittels der Ulbrichtkugel wurde so angepasst, dass das Kamerasignal voll ausgesteuert ist, was bei den Broadcastkameras bei einer Belichtungsstärke von 1250 Lux der Fall ist. Die Consumerkameras waren bei einer Belichtungsstärke von 5000 Lux ausgesteuert. Die Rohdaten der Messung finden sich in Kapitel B.7 im Anhang. Abbildung 5.7 zeigt den schematischen Aufbau des Versuchs.

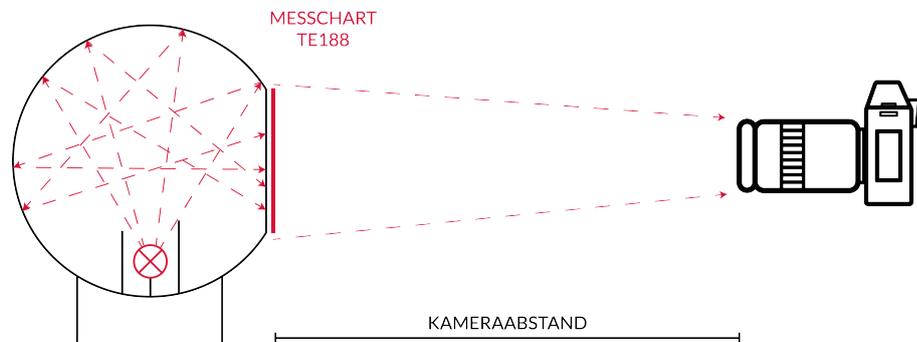


Abbildung 5.7.: Blockschaltbild der zweiten Messung aus der sechsten Messreihe

6. Auswertung

Um die in Kapitel 5 erklärten Messreihen auszuwerten, wird in diesem Kapitel in drei Schritten vorgegangen. Zunächst werden die Messgeräte verifiziert, um zu klären, ob und an welchen Stellen Messungenauigkeiten und Abweichungen zu erwarten sind. Im zweiten Schritt werden die Luminanz- und Spektraleigenschaften ausgewertet, die in den Messversuchen ermittelt wurden. Abschließend werden die ermittelten Daten interpretiert und in einen Zusammenhang mit HDR-WCG-Produktionen und den Sichtbarkeitsgrenzen des HVS gebracht.

6.1. Verifikation der Messgeräte

Aus den vier Messreihen zur Verifikation der Messinstrumente lassen sich Messergebnisse zur Genauigkeit der Messbedingungen ableiten. Im Folgenden werden die Messreihen genauer analysiert. Dabei wird zunächst die Lichtquelle auf ihre Linearität zur Beleuchtungsdichte überprüft und anschließend die Farbkonstanz des Leuchtmittels verifiziert. In den darauffolgenden Messungen werden die beiden Messcharts mit den Herstellerangaben verglichen, um zu schauen, ob die Messwerte der verwendeten Messcharts aussagekräftig sind.

6.1.1. Auswertung 1: Verifikation der Linearität der Lichtquelle

Bei der ersten Messreihe zeigt sich, dass die gemessene Leuchtdichte des Spektrometers bei allen vier Messungen annähernd linear zur Lichtstärke der Lichtquelle in der Ulbrichtkugel verläuft. Abbildung 6.1 zeigt das Verhältnis zwischen der gemessenen Leuchtdichte des Spektrometers und der gemessenen Lichtstärke der Photodiode in der Ulbrichtkugel.

Die ideale Lichtquelle I aus der Grafik lässt sich mathematisch ermitteln. Sie errechnet sich, indem jeder gemessene Wert für die Lichtstärke der Ulbrichtkugel L auf den Wertebereich von 0 bis 1 normiert, also durch den Maximalwert $\max(L)$ geteilt und anschließend mit dem Maximalwert des Spektrometers $\max(S)$ multipliziert wird. Dadurch wird jeder Wert auf den Wertebereich der Spektrometermessung linearisiert. Es gilt:

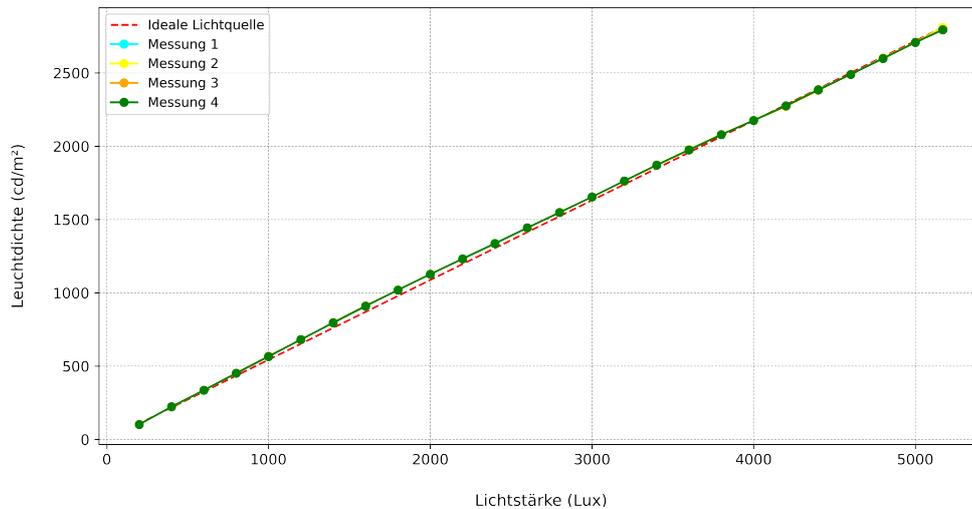


Abbildung 6.1.: gemessene Leuchtdichte des Spektrometers gegenüber der emittierten Lichtstärke der Ulbrichtkugel

$$I_n = \frac{L_n}{\max(L)} \cdot \max(S) \quad (6.1)$$

Sie wurde durch die folgende Notation in Python ermittelt. Der gesamte Quellcode findet sich im Anhang D.1.

```
1 ideal_lightsource = [x / max(measured_luminance_luxmeter_lin) *
    max(measured_luminance_spektrometer_lin_mean) for x in
    measured_luminance_luxmeter_lin]
```

Listing 6.1: Ideale Lichtquelle ermitteln

Abbildung 6.1 zeigt, dass es eine leichte Abweichung der gemessenen Daten zu einer idealen Lichtquelle gibt. Die maximale absolute Abweichung der Messreihe liegt bei 41,32 cd/m² bei einer Belichtungsstärke von 1600 Lux. Die durchschnittliche Abweichung über die gesamte Messreihe liegt bei 15,73 cd/m². Die relative Abweichung der durchschnittlich gemessenen Leuchtdichte des Spektrometers steigt, je dunkler das Leuchtmittel ist, auf einen Maximalwert von 6,08% und liegt durchschnittlich bei einer Abweichung von 2,23%.

Generell zeigt sich, dass die relative Genauigkeit der Messwerte in den hellen Bereichen der Ulbrichtkugel steigt. Dieser Effekt lässt darauf schließen, dass die absolute Schwankung über die gesamte Messreihe weitestgehend konstant bleibt, aber bei geringeren Leuchtdichten stärker ins Gewicht fällt.

Aus den absoluten Messwerten und Abbildung 6.1 ist zudem zu sehen, dass es zu einer leichten Abweichung der Linearität in den mittleren Helligkeitsbereichen der Ulbrichtkugel kommt, die jedoch relativ zur Intensität der Lichtquelle nur schwach

ins Gewicht fällt. In der folgenden Auswertung wird daher von einem linearen Verhältnis zwischen gemessenen Lichtintensitäten und der emittierten Lichtstärke der Ulbrichtkugel ausgegangen.

In der zweiten Messung wurde die Ulbrichtkugel in logarithmischen Abständen vermessen. Dadurch werden die relativen Fehler in den dunklen Bereiche stärker abgebildet und gleichzeitig die Linearität zwischen Lichtquelle und Spektromettermessung bei logarithmischen Helligkeitsabständen gezeigt. Abbildung 6.2 zeigt das Verhältnis der gemessenen Leuchtdichte des Spektrometers zu der gemessenen Lichtstärke der Photodiode in der Ulbrichtkugel in logarithmischen Abständen zwischen den Messwerten.

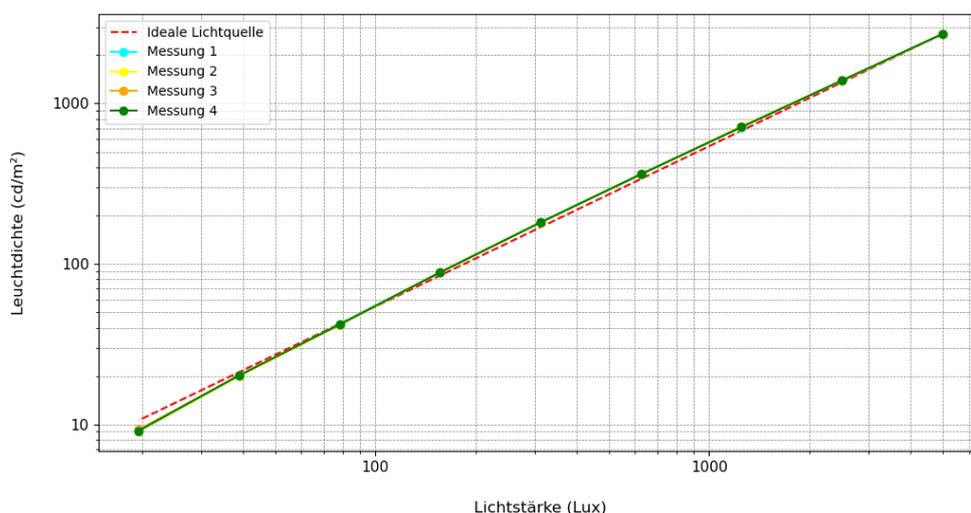


Abbildung 6.2.: gemessene Leuchtdichte des Spektrometers gegenüber der emittierten Lichtstärke der Ulbrichtkugel

Die Berechnung der idealen Lichtquelle in Abbildung 6.2 entspricht der Berechnung aus Formel 6.1. Aus den Daten ergibt sich für die zweite Messung eine maximale absolute Abweichung von $37,73 \text{ cd/m}^2$ (bei 1250 Lux) bei einer durchschnittlichen absoluten Abweichung über die gesamte Messreihe von $12,24 \text{ cd/m}^2$.

Die relative Abweichung steigt auch bei logarithmischen Messabständen in den dunklen Bereichen an. Die maximale relative Abweichung liegt bei $13,52\%$ bei einer durchschnittlichen relativen Abweichung von $5,21\%$. Für alle folgenden Messungen gilt es demnach zu beachten, dass es in den sehr dunklen Messbereichen zu einer leichten Abweichung in den Messergebnissen kommen kann. Dies gilt vor allem bei allen Messungen, die in logarithmischen Helligkeitsabständen mit der Ulbrichtkugel vermessen werden.

6.1.2. Auswertung 2: Verifikation der Farbkonstanz der Lichtquelle

In der zweiten Messreihe zur Kontinuität der Farbtemperatur bei sich ändernder Beleuchtungsstärke zeigt sich, dass die Farbtemperatur weitestgehend konstant bleibt, jedoch mit steigender Belichtungsstärke leicht abfällt. Abbildung 6.3 zeigt den Verlauf der Farbtemperatur gegenüber der Belichtungsstärke des Leuchtmittels der Ulbrichtkugel.

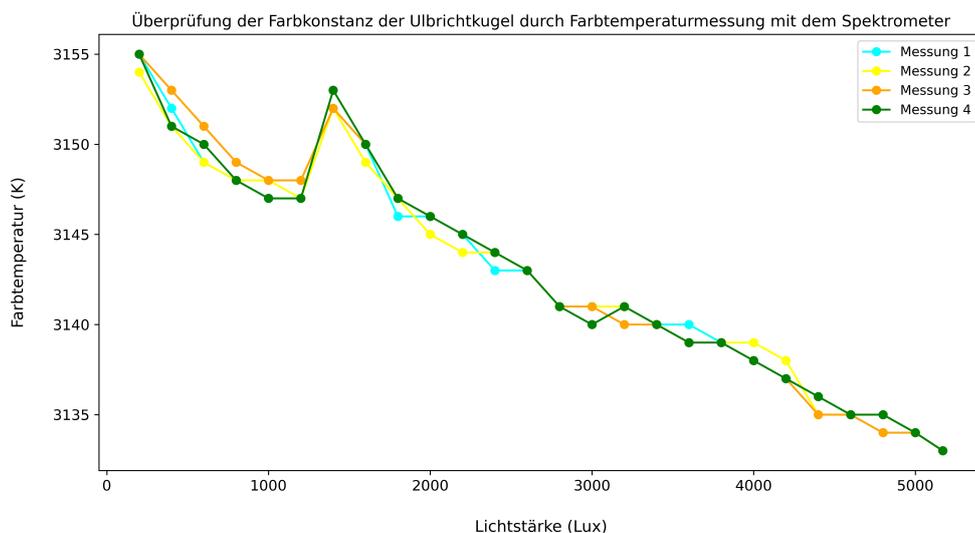


Abbildung 6.3.: gemessene Farbtemperatur des Spektrometers gegenüber der emittierten Lichtstärke der Ulbrichtkugel

Auf Grund der Skalierung von Abbildung 6.3 wirkt der Einfluss der Farbtemperatur relativ stark, jedoch schwankt die Farbtemperatur mit einem Maximalwert von 3155 Kelvin und einem Minimalwert von 3133 Kelvin relativ schwach. Etwas deutlicher wird der Einfluss in den sehr dunklen Bereichen der Ulbrichtkugel. Abbildung 6.4 zeigt den Verlauf der gemessenen Farbtemperatur des Spektrometers in logarithmischen Helligkeitsabständen der Ulbrichtkugel.

Es wird deutlich, dass in den sehr dunklen Helligkeitsbereichen die Farbtemperatur stärker abfällt und zwar auf einen Minimalwert von 3102 Kelvin bei einer Beleuchtungsstärke von $19,53 \text{ cd/m}^2$. Der Maximalwert für die logarithmischen Helligkeitsabstände liegt bei 3161 Kelvin. Das entspricht einem Abfall der Farbtemperatur von 59 Kelvin über den gesamten Helligkeitsbereich.

Für die Auswertung der Kameradaten vor allem bei den Farbmessungen gilt demnach zu beachten, dass es in den sehr dunklen Bereichen des Leuchtmittels zu leichten Abfällen in der Farbtemperatur kommt. Anhang D.2 zeigt den vollständigen Quellcode der Auswertung.

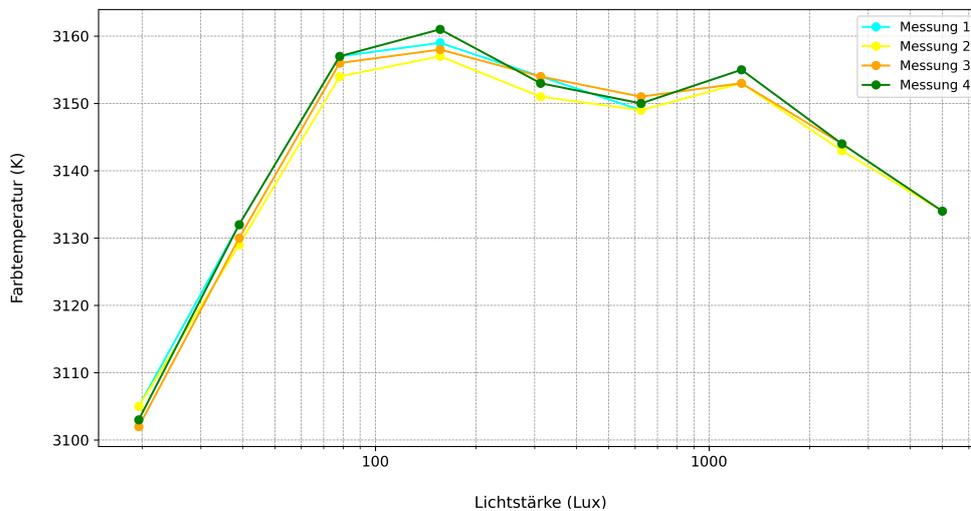


Abbildung 6.4.: gemessene Farbtemperatur des Spektrometers gegenüber der emittierten Lichtstärke der Ulbrichtkugel

6.1.3. Auswertung 3: Verifikation des Messcharts TE223

Das Messchart TE223 verfügt über 13 Lumapatches, die mit unterschiedlichen optischen Dichten des Herstellers angegeben sind. Vergleicht man die optische Dichte des Herstellers A.1 mit der gemessenen optischen Dichte des verwendeten Testcharts, lässt sich das Testchart für die Messung verifizieren.

Die optische Dichte D ist ein Maß für die Stärke der Absorption eines Materials. Sie lässt sich über die Intensität des Lichts vor I_0 und nach dem Durchgang durch das Testchart I ermitteln [34] [17]. Es gilt:

$$D = -\log_{10} \left(\frac{I}{I_0} \right) \quad (6.2)$$

Vergleicht man die gemessenen optischen Dichten mit den optischen Dichten aus den Herstellerangaben ergibt sich folgender Verlauf.

Abbildung 6.5 zeigt den Verlauf des vermessenen TE223 im Vergleich zu den Herstellerangaben. Die Kurve zeigt, dass das verwendete Testchart sehr nah an den Herstellerangaben liegt. Die leichten Abweichungen lassen sich durch die Positionierung und damit verbundenen Schwankungen durch das Spektrometer erklären, können jedoch für die folgenden Auswertungen vernachlässigt werden. Der Quellcode für die Berechnung der optischen Dichte findet sich im Anhang D.3.

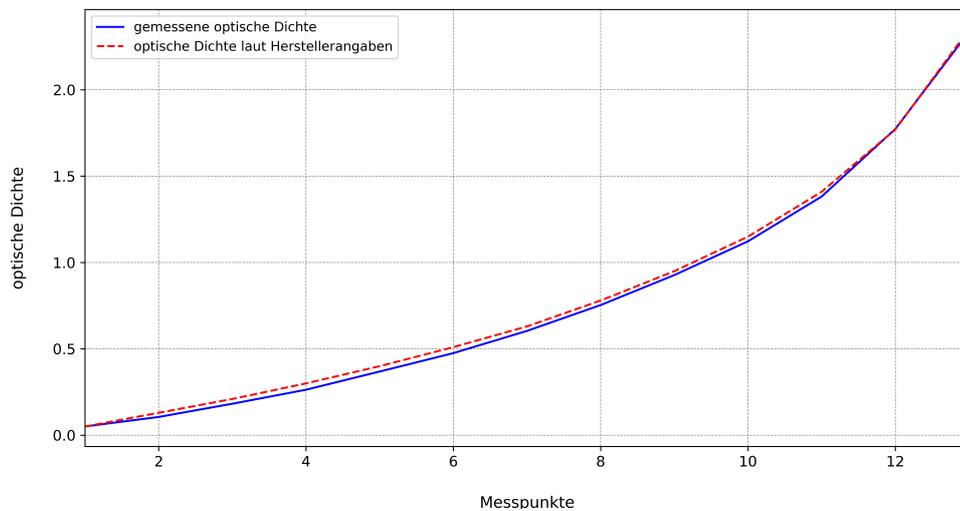


Abbildung 6.5.: gemessene optische Dichte des TE223 im Vergleich zu den Herstellerangaben

6.1.4. Auswertung 4: Verifikation des Messcharts TE188

Das Messchart TE188 der Firma Image Engineering ist ein Colorrender Testchart, welches aus 24 Patches besteht. Es teilt sich in zwölf für das Produktionsumfeld kritische Farbpatches, sechs Primär- und Sekundärfarbenpatches (R, G, B, Y, M, C) und sechs Graustufenpatches auf. In Tabelle A.3 sind die Referenzdaten vom Hersteller angegeben. Diese werden verwendet, um das Testchart zu verifizieren. Dazu wurden mit dem Spektrometer die XYZ- sowie xyY Farbpositionen gemessen. Der genaue Messaufbau findet sich in Kapitel 5. Tabelle B.4 zeigt die Messwerte des Spektrometers. Anschließend werden die Farbabstände der Messergebnisse mit den Referenzdaten des Herstellers im La^*b^* -Farbraum miteinander verglichen. Der La^*b^* Farbraum ist ein von der CIE definierter Farbraum, der die Farbwahrnehmung des Menschen in einem dreidimensionalen Raum abbildet, in dem die Farbwerte perceptiv gleichabständig zueinander sind. Dadurch kann man über die Vektorabstände eine Aussage über die Farbtreue der Messwerte zueinander treffen. Die Vektorabstände der Farbpositionen zueinander werden als ΔE definiert. Sie berechnen sich folgender Formel [36]:

$$\Delta E = \sqrt{\left(\frac{\Delta L^*}{k_L S_L}\right)^2 + \left(\frac{\Delta C^*}{k_C S_C}\right)^2 + \left(\frac{\Delta H^*}{k_H S_H}\right)^2} \quad (6.3)$$

Die Formel ist nach dem neusten Standard der CIE definiert und berücksichtigt die Helligkeit ΔL , die Farbintensität ΔC und den Farbton ΔH . Diese berechnen sich wie folgt:

$$\Delta L'^* = L_2^* - L_1^* \quad (6.4)$$

$$\Delta C'^* = C_2^* - C_1^* \quad (6.5)$$

$$\Delta H'^* = \sqrt{(a_2^* - a_1^*)^2 + (b_2^* - b_1^*)^2} - (\Delta C'^*)^2 \quad (6.6)$$

Die Formel für den Chromawert C^* ermittelt sich durch:

$$C^* = \sqrt{a^{*2} + b^{*2}} \quad (6.7)$$

In der Grundformel finden sich zudem mit k und S Gewichtung- und Skalierungsfaktoren, die für die Gewichtung nichtlinearer Zusammenhänge der menschlichen Wahrnehmung verwendet werden. Für die Berechnung wurden sie gemäß des CIE-Standards implementiert [36]. In Anhang D.4 findet sich die Berechnung für die Verifikation des TE188 Messcharts. Für die Transformation kam dabei die Colour-Bibliothek in Python zum Einsatz. Aus den Berechnungen ergibt sich folgende Tabelle für die $L^*a^*b^*$ und ΔE Werte der Messung zu den Referenzdaten:

L_Referenz	a*_Referenz	b*_Referenz	L_Messung	a*_Messung	b*_Messung	Delta E
100.0	35.171217	28.103926	100.0	45.710799	33.387309	3.796511
100.0	25.327244	22.060404	100.0	24.237194	22.646925	0.888837
100.0	-14.322470	-33.347529	100.0	-3.831140	-30.280735	7.300002
100.0	-77.037868	76.270737	100.0	-74.123097	75.179635	0.637285
100.0	22.318489	-40.042124	100.0	24.340298	-26.504038	8.467808
100.0	-43.703490	-4.740084	100.0	-38.099557	-8.695701	3.480449
100.0	52.275529	89.755738	100.0	47.267513	89.341405	2.210304
100.0	8.169642	-95.613698	100.0	4.967394	-88.832573	0.746200
100.0	72.095372	30.573099	100.0	65.900503	38.410817	4.941193
100.0	99.709082	-125.914584	100.0	102.942078	-114.811852	3.831014
100.0	134.790187	100.612074	100.0	137.341109	109.388624	2.023278
100.0	-12.040264	116.190161	100.0	-13.822455	121.515967	1.190548
100.0	143.992127	-62.063920	100.0	131.959250	-41.167024	5.502009
100.0	-28.970228	-33.110557	100.0	-29.048726	-26.815636	4.047764
100.0	0.007728	0.015779	100.0	-1.161103	4.388057	2.053593
100.0	-0.352607	0.758645	100.0	-0.959230	4.884088	1.016552
100.0	-0.474476	1.170807	100.0	-0.978738	5.066089	4.442571
100.0	-0.460295	1.261868	100.0	-0.862719	5.233891	2.729859
100.0	-0.470736	1.852478	100.0	-1.344777	5.832278	4.258083

Tabelle 6.1.: Gemessene Farbpositionen der Farbpatches und des Weißpatches des TE188

Die Wertetabelle zeigt, dass es für einen Großteil der Farben einen Farbunterschied gibt, der durchaus von dem HVS wahrgenommen werden kann. Zwischen den Messwerten und den Referenzen gibt es Farbunterschiede zwischen 0,63 und

8,47. Farbunterschiede bis zu einem ΔE von 1 können vom HVS nicht wahrgenommen werden. Ein Teil der Farben bewegt sich demnach in einem Farbunterschiedsbereich, der von einem durchschnittlichen Menschen wahrgenommen werden kann. Die Unterschiede erklären sich vor allem daraus, dass die Referenzdaten bei einer Weißpunktbeleuchtung von D65 aufgezeichnet wurden. Bei dem Messaufbau wurde auf Grund der Farbtemperatur der Lichtquelle der Referenzweißpunkt durch das weiße Patch (Patch 19) des Farbcharts verwendet und weißpunkttransformiert. Die Transformation wurde auf Grundlage der Bradford-Methode durchgeführt.

Für die Farbmessungen gilt es zu beachten, dass auf Grund der Farbunterschiede mit leichten Ungenauigkeiten in der Auswertbarkeit der Messdaten zu rechnen ist. Gleichwohl kann davon ausgegangen werden, dass große Fehler in der Messabweichung durch ein fehlerhaftes Messchart ausgeschlossen werden können. Da die Kameras unter den gleichen Bedingungen wie die Farbmessung vermessen werden, wird im Folgenden mit den gemessenen Daten weitergearbeitet. In Abbildung 6.6 werden die Abweichungen der Farbpatches im dreidimensionalen Raum dargestellt.

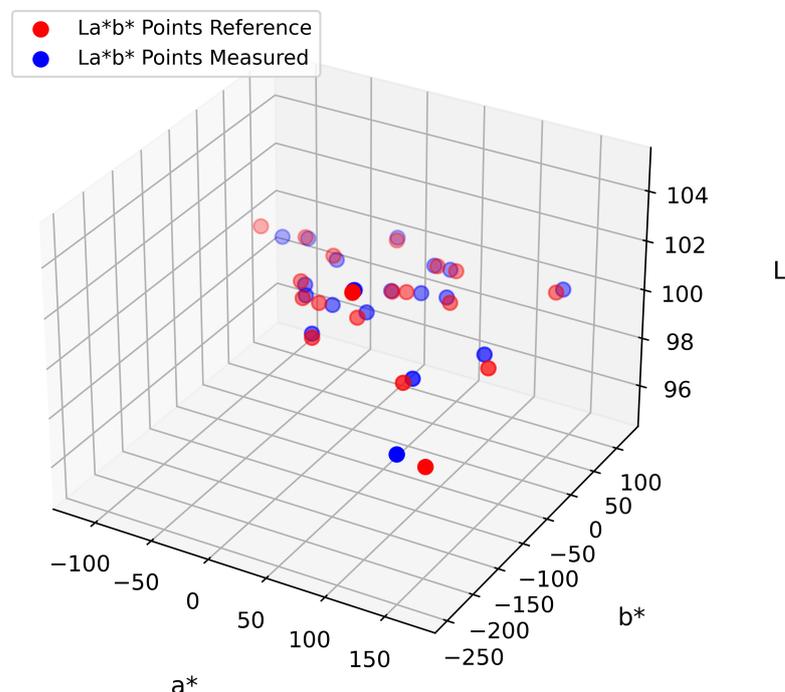


Abbildung 6.6.: Die gemessenen Farbpositionen des Farbpatches im Vergleich zur Referenz

6.2. Kameraanalyse

Rohdaten einer Kamera liegen in der szenenlinearen Domäne vor. Dadurch verhalten sich die Signalwerte der Kamera linear zu Beleuchtungsichte des auf den Sensor fallenden Lichts. Für Kamerasysteme, die einen Rohdatenzugriff zulassen, ermöglicht dies, die Signaldaten der Kamera detailliert zu analysieren. In dieser Sektion werden die Sony A7IV, die Sony A7III, die Lumix GH5s, die Phantom 4 Pro und die GoPro deshalb genauer betrachtet.

6.2.1. Auswertung 5: Gammakurve und Full-Well-Capacity

Die Gammakurve eines Kamerasystems gibt die Signalstärke der Pixelwerte im Verhältnis zu der Beleuchtungsichte des Lichts auf dem Kamerasensor an. Sie zeigt demnach die Verarbeitung des Lichts durch den Kamerasensor. Um zu verifizieren, dass die Kamerasensoren linear verlaufen, wurde gemäß des Messaufbaus aus Abschnitt 5.4.2 die Gammakurve der Kamerasysteme ermittelt.

Darüber hinaus lässt sich aus der Kombination aus bekannten Signalwerten und Luminanzen die Full-Well-Capacity, also die Sättigung der einzelnen Kamerapixel ermitteln. Die Sättigung ist bei linearen Verhältnissen zwischen Beleuchtungsichte und Signalwert dann erreicht, wenn der Signalwert 1 ist. Da sich die Sättigung einzelner Pixel leicht unterscheiden kann, wird in dieser Auswertung der Durchschnitt über alle Pixel herangezogen, um auf den durchschnittlichen Wert für die Sensorkapazität zu schließen. Eine genauere Betrachtung der einzelnen Pixeleigenschaften wird in Kapitel 6.2.3 und Kapitel 6.2.4 erläutert.

Durch die Signalwerte des TE223 durch den Kamerasensor und das Ermitteln von Messwerten mit dem Spektrometer, erhält man eine Reihe an Wertpaaren. Diese Wertpaare werden verwendet, um die Gammakurve abzubilden. Je höher die Anzahl der Wertpaare, desto genauer lässt sich die Gammakurve abbilden. Es gilt zu beachten, dass die Wertpaare sich bestenfalls gleichabständig über die gesamte Kapazität des Sensors verteilen, um den Fehler zu minimieren. Durch die Verwendung des TE223, in dem die optische Dichte der Lumapatches logarithmisch steigt, werden die mittleren und dunklen Bereiche der Gammakurve in dieser Messung stärker berücksichtigt. Dies gilt es bei der Auswertung zu beachten.

Neben der Gammakurve lässt sich aus den Messwerten die durchschnittliche Kapazität der Sensorpixel ermitteln. Die Sensorkapazität ist ein Maß für den maximalen Lichteinfall, den eine Kamera verarbeiten kann, ohne ins Clipping zu geraten. Je höher die Sensorkapazität ist, desto mehr Licht kann ein Sensor verarbeiten. Teilt man die gemessene Beleuchtungsichte der Lumapatches durch den Signalwert, der durch die Kamera aufgenommen wurde, erhält man die durchschnittliche Sensorkapazität dieser Sensorregion. Um lokale Schwankungen des Sensors best-

möglich auszugleichen, wurde die durchschnittliche Sensorkapazität für alle Wertpaare erhoben und anschließend der Mittelwert berechnet. Dadurch erhält man die durchschnittliche Full-Well-Capacity für den gesamten Sensorbereich. Sie lässt sich durch die folgende Formel beschreiben, wobei n der Anzahl der Lumapatches, L_i der Beleuchtungsdichte des i -ten Lumapatches und S_i dem entsprechenden Signalwert der Kamera entspricht.

$$\text{Full-Well-Capacity} = \frac{1}{n} * \sum_{i=1}^n \frac{L_i}{S_i} \quad (6.8)$$

Im Folgenden werden die gemessenen Gammakurven aller Kameras erläutert. Für die folgenden Graphen zur Gammakurve gilt es zu beachten, dass das Spektrometer bei einer Beleuchtungsstärke von ca. 7 cd/m² durch das niedrige Helligkeitslevel keine Messergebnisse mehr liefern konnte. Alle Werte, die unter diesem Punkt liegen, sind daher aus den Werten bei einer helleren Belichtung errechnet worden, indem der Luminanzwert durch den Faktor 2 geteilt wurde. Die genauen Messergebnisse und errechneten Messdaten finden sich im Anhang B.5 und Anhang C.1. Der für die Berechnung verwendete Quellcode findet sich in Anhang D.5. In den sehr dunklen Bereichen kommt es zu einem Abfall in den Messwerten. Diese lassen sich durch die Black-Level-Subtraction erklären. Der Schwarzwert für die Grafiken wurde aus den Dunkelbildern ermittelt und weicht damit von dem tatsächlichen Blacklevel ab. Diese Abweichung verursacht den Abfall in der Kurve. Sie ließe sich durch Anpassung des Schwarzwerts ausgleichen. Da sie jedoch außerhalb der Messwerte liegt und damit nur angenähert werden können, werden diese Bereiche für die Auswertung unberücksichtigt gelassen.

Oberhalb der Sensorkapazität kommt es zudem zu Wertpaaren, die bei einem Signalwert von 1 unterschiedlich gemessenen Helligkeitswerte haben. Dabei handelt es sich um die Clippinggrenze des Sensors.

Abbildung 6.7 zeigt beispielhaft den Verlauf für die Sony A7IV. Alle anderen Abbildungen finden sich auf Grund der Menge im Anhang.

Abbildung C.1 und 6.7 zeigt die gemessene Gammakurve und die ermittelte Full-Well-Capacity der Sony A7IV. Die Gammakurve zeigt einen linearen Verlauf. Die Sensorkapazität der Kamera bei Blende F8.0, ISO = 100 und einer Belichtungszeit von 1/60 Sekunden ist bei einer Beleuchtungsdichte der Ulbrichtkugel von 8246.92 cd/m² erreicht.

Abbildung C.2 zeigt die gemessene Gammakurve und die ermittelte Full-Well-Capacity der Sony A7III. Die Gammakurve zeigt einen linearen Verlauf. Die Sensorkapazität der Kamera bei Blende F8.0, ISO = 100 und einer Belichtungszeit von 1/60 Sekunden ist bei einer Beleuchtungsdichte der Ulbrichtkugel von 7647,61 cd/m² erreicht.

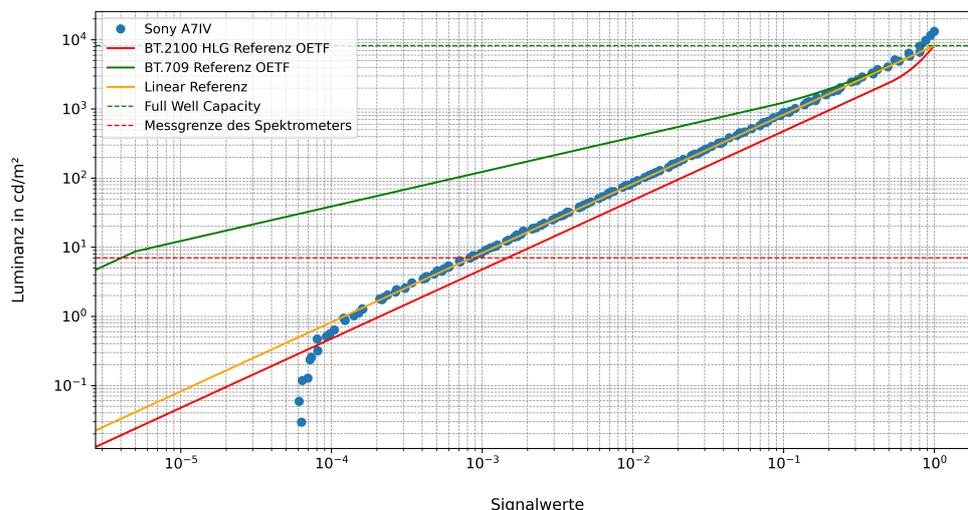


Abbildung 6.7.: Gammakurve und Sättigungskapazität der Sony A7IV

Abbildung C.3 zeigt die gemessene Gammakurve und die ermittelte Full-Well-Capacity der Panasonic Lumix GH5s. Die Gammakurve zeigt einen linearen Verlauf. Die Sensorkapazität der Kamera bei Blende F8.0, ISO = 100 und einer Belichtungszeit von 1/60 Sekunden ist bei einer Beleuchtungsichte der Ulbrichtkugel von 8449,34 cd/m^2 erreicht.

Abbildung C.4 zeigt die gemessene Gammakurve und die ermittelte Full-Well-Capacity der Phantom 4 Pro. Die Gammakurve zeigt einen linearen Verlauf. Die Sensorkapazität der Kamera bei Blende F8.0, ISO = 100 und einer Belichtungszeit von 1/60 Sekunden ist bei einer Beleuchtungsichte der Ulbrichtkugel von 6551,54 cd/m^2 erreicht.

Abbildung C.5 zeigt die gemessene Gammakurve und die ermittelte Full-Well-Capacity der GoPro 10. Die Gammakurve zeigt einen linearen Verlauf. Die Sensorkapazität der Kamera bei Blende F2.5, ISO = 100 und einer Belichtungszeit von 1/125 Sekunden ist bei einer Beleuchtungsichte der Ulbrichtkugel von 2710,29 cd/m^2 erreicht. Da die GoPro 10 über eine feste Blende verfügt und die Belichtungszeit nicht auf 1/60 Sekunden eingestellt werden kann, wurden diese Einstellungen gewählt und die Helligkeitswerte der Ulbrichtkugel entsprechend modifiziert, sodass sich vergleichbare Signal- und Helligkeitswerte zu den anderen Kameras ergeben. Die genauen Messwerte finden sich im Anhang B.5

Aus den Messergebnissen geht hervor, dass die Kamerasysteme wie zu erwarten alle eine lineare Abhängigkeit zwischen Signalwerten und Beleuchtungsichte des Sensors haben. Bei den Consumerkameras, also der Sony A7IV, Sony A7III und der Panasonic Lumix GH5s, misst man sehr vergleichbare Sensorkapazitäten. Auf Grund der deutlich kleineren Sensoren erreichen die Phantom 4 Pro und die GoPro 10 etwas kleinere Maximalwerte. Die folgende Tabelle zeigt die Sensorkapazitäten

im Vergleich:

Kamera	Sensorkapazität (cd/m ²)
Sony A7IV	8246,92
Sony A7III	7647,61
Lumix GH5s	8449,34
Phantom 4 Pro	6551,54
GoPro 10	2710,29

Tabelle 6.2.: gemessene Sensorkapazität in cd/m²

Für die Broadcastkameras lässt sich ebenfalls die Gammakurve der Kamerasysteme ermitteln und der HLG Workflow verifizieren. Die Full-Well-Capacity lässt sich dadurch jedoch nicht ohne Weiteres berechnen, da durch die Transferfunktion und die Kameracharakterisierung die Bilddaten in der displaylinearen Domäne vorliegen und vorverarbeitet werden. Die Werte sind deshalb auf diese Weise nur bedingt messbar. Abbildung 6.8 zeigt die Gammakurve der Panasonic UC4000 stellvertretend für die Broadcast-Kamerasysteme. Sie zeigt den Verlauf der Werte zu der HLG-Referenzkurve zur Verifikation der Bilddaten.

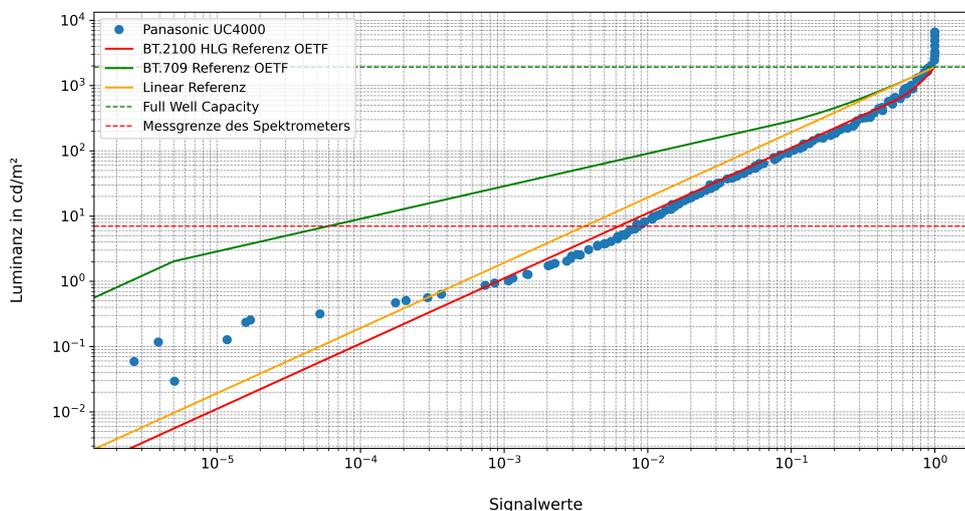


Abbildung 6.8.: Gammakurve der UC4000

Die Panasonic UC4000 und die Ikegami UHK-X700 konnten durch die Gammakurve als HLG referenziert werden. Die Grass Valley LDX135 wurde fälschlicherweise durch einen Dual-HDR-SDR Workflow im Rec.709 Workflow vermessen. Die Messwerte der Kameras entsprechen hier zwar den gemessenen Leuchtdichten der Kamerasysteme, die Signalwerte sind aber in einem vorverarbeiteten Zustand und entsprechen demnach nicht eindeutig den Sensorwerten. Die Grafik muss in diesem Kontext betrachtet werden.

6.2.2. Auswertung 6: Vignette

Die Vignette ist ein Qualitätskriterium für Kamera-Objektiv-Kombinationen und kann Einfluss auf Messungen nehmen, die wie bei den Testcharts TE223 und TE188 in unterschiedlichen Bereichen des Sensors durchgeführt werden. Die Vignette tritt durch eine Abschattung an den Randbereichen des Sensors auf. Sie entsteht meist durch die Konstruktionsweise der Kameraobjektive. In diesem Abschnitt wird die Vignette als Qualitätskriterium genauer betrachtet.

Eine Vignette tritt für gewöhnlich umso stärker auf, je weiter der Signalwert vom Sensormittelpunkt entfernt liegt. Aus den Signalwerten von Messung 5.4.2 lässt sich die Vignette ermitteln, indem jeder Pixel auf Grund seiner Entfernung von der Sensormitte in Pixelgruppen eingeteilt wird. Für die Einteilung wird das Bild in 100 Abschnitte unterteilt, die jeweils eine Toleranz von 1% aufweisen. Dadurch lassen sich alle Pixel in 100 Gruppen entsprechend ihrer Distanz zur Bildmitte einteilen. Abbildung 6.9 zeigt die Berechnung der Pixelabstände mittels des Satzes des Pythagoras und die Ränder der Pixelgruppen (hier durch Kreise angedeutet).

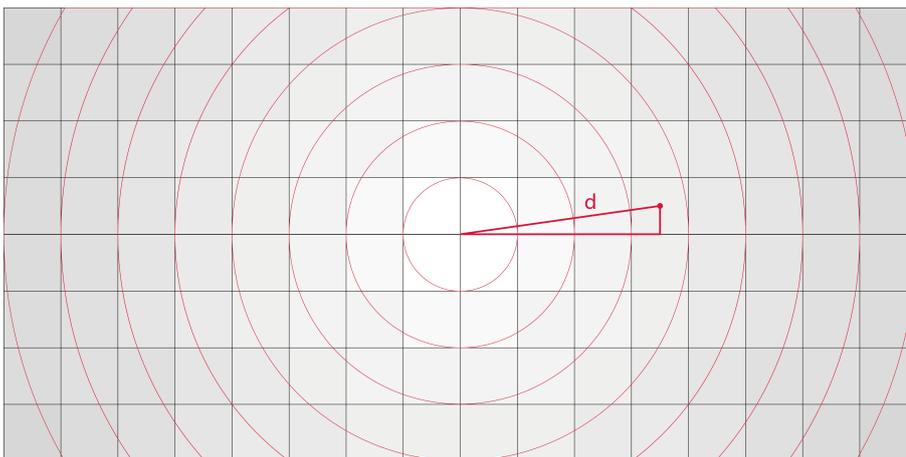


Abbildung 6.9.: Beispielbild für Vignettierung und Berechnung der Entfernung einzelner Pixel

Die Berechnung der Entfernung jedes Pixels zur Sensormitte wurde wie folgt umgesetzt, wobei height der Höhe des Bildes und width der Breite des Bildes entspricht.

```

1   Y, X = np.ogrid[:height, :width]
2   distance_from_center = np.sqrt((X - center_width)**2 + (Y -
    center_height)**2)

```

Listing 6.2: Verifikation der Ulbrichtkugel (Luminanz)

Nimmt man den Durchschnitt der Signalwerte aller Pixel innerhalb einer Pixelgruppe, so erhält man eine Reihe an Signalwerten, die in Relation zu ihrem Abstand zur Sensormitte stehen. Visualisiert man diese Wertreihe, so lässt sich die Vignettierung eines Signals darstellen. Auf Grund der Vielzahl an Grafiken wird für die Visua-

lisierung die Vignette der Sony A7IV gezeigt (6.10). Alle weiteren Visualisierungen finden sich im Anhang.

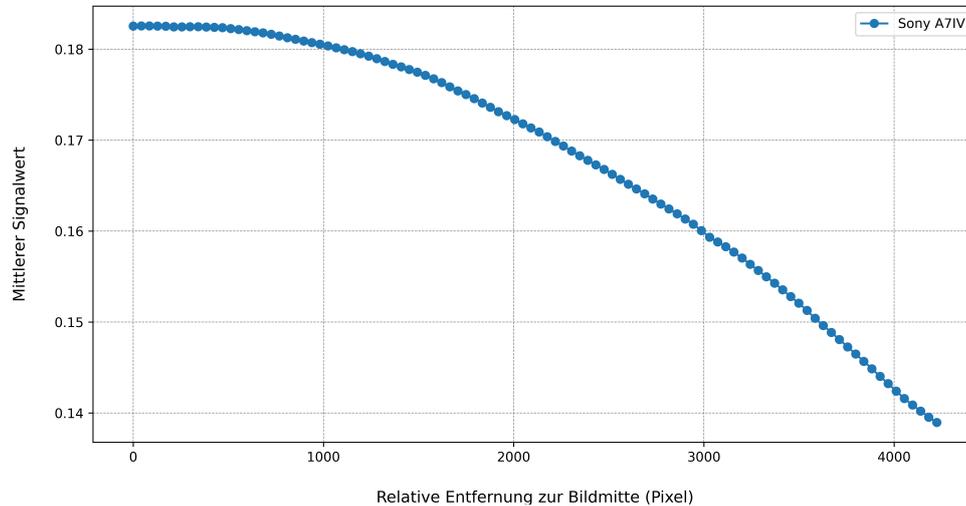


Abbildung 6.10.: Vignette der Sony A7IV

Abbildung C.6 zeigt ebenfalls die Vignette der Sony A7IV. Man sieht einen gleichmäßigen Signalwertverlust in Richtung der Bildränder. Der maximale Signalwert in der Sensormitte liegt bei 0,1826 Digital Units (DU). Der minimale Signalwert an den Sensorecken liegt bei 0,1390 DU. Das entspricht einer Abweichung von 0,0436 DU und einem relativen Signalwertverlust von 23,88%.

Abbildung C.7 zeigt die Vignette der Sony A7III. Man sieht einen gleichmäßigen Signalwertverlust in Richtung der Bildränder. Der maximale Signalwert in der Sensormitte liegt bei 0,1996 DU. Der minimale Signalwert an den Sensorecken liegt bei 0,1395 DU. Das entspricht einer Abweichung von 0,0601 DU und einem relativen Signalwertverlust von 30,13%.

Abbildung C.8 zeigt die Vignette der Panasonic Lumix GH5s. Man sieht einen gleichmäßigen Signalwertverlust in Richtung der Bildränder. Der maximale Signalwert in der Sensormitte liegt bei 0,1497 DU. Der minimale Signalwert an den Sensorecken liegt bei 0,1397 DU. Das entspricht einer Abweichung von 0,0100 DU und einem relativen Signalwertverlust von 6,69%. Für die Lumix GH5s gilt es zu beachten, dass das Sigma 18-35mm Objektiv mit dem Viltrox EF-M1 Adapter auf die Kamera adaptiert wurde. Bei dem Sigma Objektiv handelt es sich um ein APSC-Objektiv, was auf einer MFT-Kamera verwendet wird. Dadurch wird der Lichtkegel des Objektivs nicht vollständig ausgenutzt und die kritischen Randbereiche die die Vignettierung am stärksten beeinflussen abgeschnitten.

Abbildung C.9 zeigt die Vignette der Phantom 4 Pro. Man sieht einen gleichmäßigen Signalwertverlust in Richtung der Bildränder. Der maximale Signalwert in der Sensormitte liegt bei 0,2490 DU. Der minimale Signalwert an den Sensorecken liegt

bei 0,1097DU. Das entspricht einer Abweichung von 0,1393DU und einem relativen Signalwertverlust von 55,95%.

Abbildung C.10 zeigt die Vignette der GoPro10. Man sieht einen gleichmäßigen Signalwertverlust in Richtung der Bildränder. In den mittleren Abstandswerten kommt es zu einem unerwarteten Peak in den Signalwerten. Dieser Peak ist auf einen Messfehler in den Bildern zurückzuführen, da beim Framing der Signalwerte die Lichtquelle der Ulbrichtkugel mit aufgezeichnet wurde. Für alle weiteren Messungen werden die Bilder der GoPro daher zugeschnitten. Für die Messung der Vignette würde es jedoch den Signalwertverlust beeinflussen. An den Rändern sind die Effekte vernachlässigbar, weshalb der maximale und minimale Signalwert weiterhin aussagekräftig sind. Der maximale Signalwert in der Sensormitte liegt bei 0,3070 DU. Der minimale Signalwert an den Sensorecken liegt bei 0,1159 DU. Das entspricht einer Abweichung von 0,1910 DU und einem relativen Signalwertverlust von 62,23%. Tabelle 6.3 fasst die Signalwertverluste durch Bignettierung für die untersuchten Kameras zusammen.

Kamera	Signalwertverlust (%)
Sony A7IV	23,88
Sony A7III	30,13
Lumix GH5s	6,69
Phantom 4 Pro	55,95
GoPro 10	62,23

Tabelle 6.3.: gemessene Signalwertverluste durch Vignettierung der Kameras

6.2.3. Auswertung 7: Rauschen

Aus den in Messung 5.4.2 gewonnenen Dunkelbildern lässt sich das Rauschen der Kamerasensoren ermitteln und auswerten. In einer Kamera treten verschiedene Rauscheffekte auf (siehe 4). Durch die Aufzeichnung der Dunkelbilder lässt sich das Ausleserauschen von dem Signalwert trennen. In dieser Auswertung wird das auftretende Rauschen in den Kameras miteinander verglichen.

Das Rauschen ist abhängig von der Vorverstärkung des Kamerasensors, da die Verstärkung des Signals auch zu einer Verstärkung des Ausleserauschens führt. Um eine Aussage zu treffen, wie rauschanfällig eine Kamera bei unterschiedlichen Belichtungssituationen ist, wurde von den Kamerasystemen das Ausleserauschen bei allen einstellbaren Vorverstärkungen ermittelt. Die Belichtungseinstellungen bleiben dabei konstant bei einer Blende von F8 und einer Belichtungszeit von 1/60 Sekunde.

Für die Sony A7IV und A7III ergibt das einen Vorverstärkungsbereich von 50-204400 ISO wobei zu beachten ist, dass der Bereich zwischen 100-51200 auf die Vorverstärkung durch die Sensorelektronik zurückzuführen ist. Die ISO-Werte in den Be-

reichen von 50-100 werden softwarebasiert durch Absenkung des Signalpegels in der Empfindlichkeit reduziert. Bei dem ISO Bereich über 51200 wird der Signalpegel nachträglich angehoben, wodurch sich der erweiterte ISO-Bereich erklärt. Auch hier wurde beispielhaft wieder die Abbildung der Sony A7IV zur Veranschaulichung in den Text eingebunden. Alle weiteren Grafiken finden sich, ebenso wie der vollständige Quellcode im Anhang (siehe C.3 und D.9).

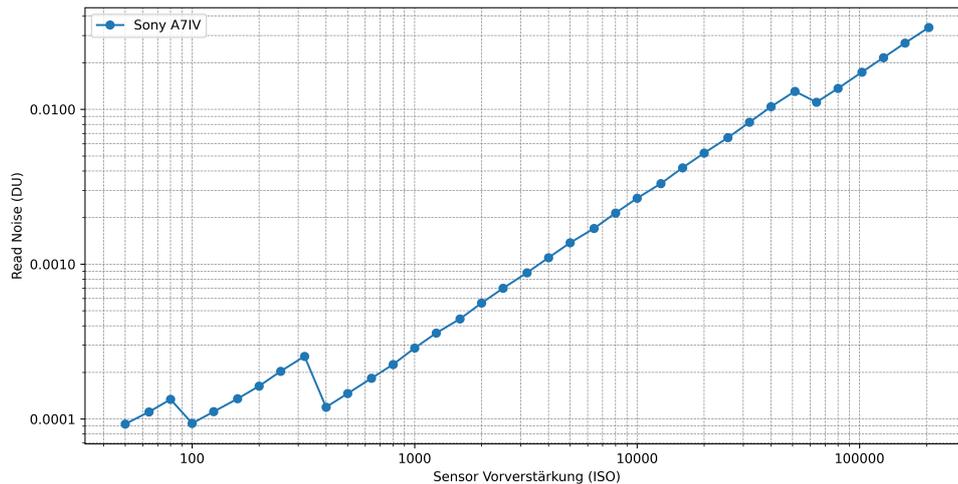


Abbildung 6.11.: Ausleserauschen der Sony A7IV

Abbildung C.11 und 6.11 zeigt das Ausleserauschen der Kamera im Verhältnis zu der Sensorverstärkung. Der Verlauf des Rauschens zeigt drei lokale Maximas. Der erste und dritte Peak sind auf die bereits angesprochene digitale Erweiterung des ISO-Bereichs zurückzuführen. Der zweite Peak bei einem ISO von 320 auf 400 deutet auf einen ISO-Schaltpunkt (Gain Step) in der Vorverstärkung des Sensors hin. Dieser entsteht immer dann, wenn die Art der Verstärkung des Sensors wechselt, beispielsweise durch verschiedene Verstärkungsschaltkreise.

Aus den Daten ergibt sich für die Sony A7IV folgendes Rauschbild. Das maximale Rauschsignal bei einer Verstärkung von 204400 liegt bei einem Signalwert von 0,0338DU. Das minimale Rauschsignal bei einem ISO von 100 liegt bei 0,0000937DU.

Die Sony A7III verfügt über den gleichen ISO-Bereich wie die Sony A7IV, jedoch liegt der ISO-Schaltpunkt hier bei einer Signalverstärkung von ISO 640. Abbildung C.12 zeigt den Verlauf des Ausleserauschs im Verhältnis zur Vorverstärkung des Sensors. Aus den Daten ergibt sich für die Sony A7III folgendes Rauschbild. Das maximale Rauschsignal bei einer Verstärkung von 204400 liegt bei einem Signalwert von 0,0210DU. Das minimale Rauschsignal mit einer Verstärkung von ISO 100 liegt bei 0,000130DU.

Die Panasonic Lumix GH5s unterstützt einen Vorverstärkungsbereich von 80 - 204400 ISO mit ebenfalls einer Maximalverstärkung des Sensors bis zu einer Vor-

verstärkung von ISO 51200 und einem Base ISO von 100. Der Verlauf des Ausleserausens aus Abbildung C.13 zeigt einen ISO-Schaltpunkt bei ISO 800. Das maximale Rauschsignal liegt bei einem Signalwert von 0,0499DU mit einer Vorverstärkung von ISO 204400. Das minimale Rauschsignal liegt bei einem Signalwert von 0,00022DU mit einer Vorverstärkung von ISO 100.

Bei der Phantom 4 Pro liegt der Vorverstärkungsbereich bei ISO 100 - ISO 12800, wobei die Verstärkung in ganzen Blendenstufen anpassbar ist. Der Base ISO liegt bei ISO 100 und hat ein Ausleserauschen von 0,00049DU. Besonders bei der Phantom 4 Pro ist, dass das Ausleserauschen bis zu einer Vorverstärkung on ISO 1600 stetig steigt und anschließend ein Plateau erreicht mit einem Signalwert von 0,00315DU. Ein höherer ISO nimmt demnach keinen Einfluss mehr auf das Ausleserauschen. Das lässt vermuten, dass auf Grund der internen Signalverarbeitung der Daten ab einer Vorverstärkung von ISO 1600 keine analoge Steigerung der Sensorempfindlichkeit mehr durchgeführt wird, sondern die Daten für höhere Signalwerte digital ermittelt werden. Abbildung C.14 zeigt den Verlauf des Ausleserausens im Verhältnis zur Vorverstärkung des Sensors.

Die GoPro verfügt über einen noch kleineren Vorverstärkungsbereich von ISO 100 - ISO 3200, was auch auf den kleinen Sensor und die geringe Sättigungskapazität der Kamera zurückzuführen ist. Der Base ISO liegt bei ISO 100 und hat ein Ausleserauschen von 0,00042DU. Das maximale Rauschsignal liegt bei ISO 3200 vor mit einem Rauschwert von 0,00772DU. Der in Abbildung C.15 dargestellte Rauschverlauf zeigt ein Plateau bei den ISO Werten von ISO 200 und ISO 400.

6.2.4. Auswertung 8: Signal-Rauschabstand und Dynamikumfang

Durch die Mittelwerte der Signalbilder und die Standardabweichungen der Dunkelbilder lässt sich der SNR berechnen. Der SNR gibt die Abweichung zwischen Signalwert und Rauschen bei einer Kameraaufnahme an. Er ermittelt sich über folgende Formel wobei S dem Mittelwert aller Pixel des Signalbildes entspricht und σ_D der Standardabweichung der Dunkelbilder:

$$\text{SNR} = \frac{S}{\sigma_D} \quad (6.9)$$

Um den Abstand des SNR geeignet darstellen zu können, wird der SNR in dieser Arbeit in dB, Blendenstufen und Density Units angegeben, wobei die Auswertung vorwiegend in dB und Blendenstufen stattfindet. Die Berechnung der verschiedenen Normierungen ist über die folgenden Formeln angegeben [42]:

$$\text{SNR}_{\text{dB}} = 20 \cdot \log_{10}(\text{SNR}) \quad (6.10)$$

$$\text{SNR}_{\text{density_units}} = 10 \cdot \log_{10}(\text{SNR}) \quad (6.11)$$

$$\text{SNR}_{f_stops} = \log_2(\text{SNR}) \quad (6.12)$$

Der SNR ist ein wichtiges Qualitätskriterium für Kamerasysteme und lässt eine Aussage darüber zu, wie rauscharm ein Bild bei bestimmten Belichtungssituationen erscheint. In den folgenden Auswertungen wurde der SNR für alle Vorverstärkungen der Kamerasysteme ermittelt. Bei den Messungen wurden die Signalwerte nicht vollständig ausgereizt, was zu einem geringfügig niedrigeren absoluten SNR führt. Der Grund dafür liegt in der Auswertbarkeit der einzelnen Kameras zueinander. Da die Kamerasysteme unterschiedlich empfindlich auf Licht reagieren, aber die gleichen „Belichtungssituationen“ simuliert werden sollten, erlaubt diese Messung eine Aussage zwischen den einzelnen Kamerasystemen. Der maximale SNR würde in der Praxis bei Vollaussteuerung des Sensors geringfügig höher ausfallen. Um den maximalen SNR zu ermitteln, wird er daher mathematisch errechnet, indem der SNR durch den maximalen Signalwert geteilt wird, was einer Normierung auf den Signalbereich von 0 - 1 entspricht. Es gilt:

$$\text{SNR}_{\text{max}} = \max\left(\frac{\text{SNR}}{S}\right) \quad (6.13)$$

Da das auf den Sensor fallende Licht die Temperatur des Sensors und damit das Temperaturrauschen beeinflussen würde, entspricht diese Normierung einer Annäherung an den maximalen SNR. Um den tatsächlichen Wert zu ermitteln, sollte schon bei der Messung die Vollaussteuerung des Sensors beachtet werden. Im Folgenden werden die Annäherungen an den maximalen SNR verglichen. Der vollständige Quellcode zur Auswertung findet sich im Anhang D.9

Die Sony A7IV verfügt über den höchsten SNR aller fünf gemessenen Consumerkameras mit einem errechneten Maximalwert von 80,55dB und einem gemessenen Maximalwert von 71,04dB bei einem Signalwert von 0,33 DU. Abbildung C.16 zeigt den Verlauf über alle verfügbaren Vorverstärkungen der Sony A7IV. Auch bei dem SNR zeigt sich die Charakteristik, dass bei einem ISO von 400 ein Umschaltprozess bei der Sensorverstärkung stattfinden muss (vgl. Kapitel 6.2.3).

Bei der Sony A7III liegt der gleiche Vorverstärkungsbereich vor. Das Vorgängermodell der Sony A7IV schafft es auf einen etwas geringeren maximalen SNR von 77,76dB bei einem gemessenen Wert von 69,12dB mit einem Signalwert von 0,37 DU. Abbildung C.17 zeigt den Verlauf des SNR über alle verfügbaren Vorverstärkungen der Sony A7III. Der spätere Gain Step der Kamera bei ISO 640 sorgt dafür, dass der SNR in den dunkleren Belichtungssituationen etwas größer ist. Das macht

die Sony A7III zu der Kamera mit dem höchsten SNR in den dunklen Belichtungssituationen. Der minimale SNR bei einer Vorverstärkung von ISO 204400 liegt bei 17,37dB.

Die Panasonic Lumix GH5s unterstützt in den hellen Bereichen zwei ISO-Steps weniger als die Sony-Kameras. Ihr gemessener SNR bei einer Vorverstärkung von ISO 80 liegt bei 56,33dB. Der zugehörige Signalwert beträgt 0,14 DU. Das ergibt einen errechneten Maximalwert von 72,63dB. Abbildung C.18 zeigt den Verlauf der Panasonic Lumix GH5s. Der Gain Step der Lumix GH5s liegt bei einer Vorverstärkung von ISO 800. Der späte ISO-Schaltspunkt sorgt für einen verbesserten SNR in den dunklen Belichtungssituationen. Abbildung 6.12 zeigt, wie sich der Kurvenverlauf der GH5s dem Kurvenverlauf der Sony A7IV annähert. Dadurch kommt sie der Sony A7IV trotz des deutlich kleineren Sensors in den mittleren bis dunklen Belichtungssituationen sehr nah, was den SNR angeht.

Die Phantom 4 Pro und die GoPro 10 zeigen einen sehr ähnlichen Kurvenverlauf trotz einer deutlich geringeren Sättigungskapazität der GoPro 10, was für die Rauschunempfindlichkeit des Sensors spricht.

Die Phantom 4 Pro erreicht einen maximalen SNR von 64,93dB bei einer Vorverstärkung von ISO 100. Der gemessene Wert liegt bei 53,36dB bei einem Signalwert von 0,23 DU. Der Signalwert bei der maximalen Vorverstärkung von ISO 12800 liegt bei 0,11 DU, was einem SNR von 6,86dB entspricht. Abbildung C.19 zeigt den Verlauf des SNR und des errechneten maximalen SNR der Phantom 4 Pro.

Die GoPro 10 erreicht bei der gleichen Vorverstärkung von ISO 100 einen etwas höheren maximalen SNR von 67,45dB. Der gemessene Wert für ISO 100 liegt bei einem Signalwert von 0,27DU und einem SNR von 56,17dB. Die maximale Vorverstärkung der GoPro 10 ist auf ISO 3200 einstellbar. Bei dieser Vorverstärkung erreicht die GoPro bei einem Signalwert von 0,24 DU einen SNR von 29,88dB.

In der folgenden Tabelle 6.4 sind die wichtigsten Messwerte zusammengefasst.

Camera	Max SNR (db)	ISO 1	SNR 1 (dB)	Signalwert 1 (DU)	ISO 2	SNR 2 (db)	Signalwert 2 (DU)
Sony A7IV	80,55	50	71,04	0,33	204400	12,56	0,14
Sony A7III	77,76	50	69,12	0,37	204400	17,37	0,16
Lumix GH5s	72,63	80	56,33	0,14	204400	6,86	0,11
Phantom 4 Pro	64,93	100	53,36	0,23	12800	16,91	0,021
GoPro 10	67,45	100	56,17	0,27	3200	29,88	0,24

Tabelle 6.4.: SNR Daten der Consumerkameras

Über den maximalen SNR lässt sich eine Aussage über die Qualität des Signals bei kontrollierten Lichtbedingungen machen. Die Tabelle zeigt, dass die Sony A7IV unter kontrollierten Bedingungen den besten SNR vorweisen kann. Betrachtet man jedoch den Verlauf der Kamerasysteme, so fällt auf, dass durch den früheren Gain-Step der Sony A7IV die Kamera in den sehr dunklen Bereichen einen etwas schwächeren SNR erreicht. Hier schneidet das Vorgängermodell, die Sony A7III etwas

besser ab. Die Panasonic Lumix GH5s liegt, wie zu erwarten, leicht darunter, was auf die Sensorgröße zurückzuführen ist. Die Phantom 4 Pro und die GoPro weisen durch den noch kleineren Sensor einen noch geringeren SNR über alle Vorverstärkungen auf. Abbildung 6.12 zeigt den Verlauf aller Kamerasysteme.

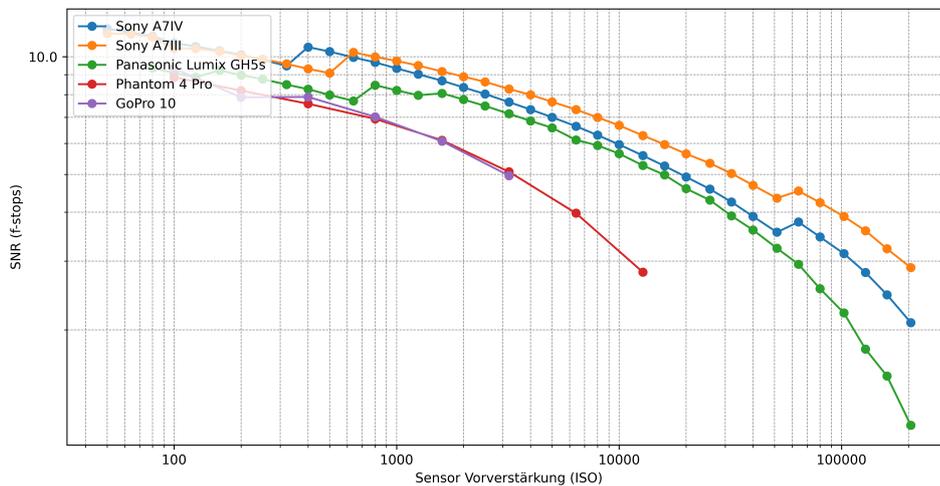


Abbildung 6.12.: Signal-Rauschabstand aller Consumerkameras

6.2.5. Auswertung 9: DSNU

Die DSNU ist eine Bewertungsmethode, um Dunkelbilder von Kamerasystemen bewerten zu können. Das Ausleserauschen in Kamerasystemen ist von verschiedenen Faktoren abhängig (vgl. Kapitel 4), wodurch Fluktuationen in dem Rauschsignal auftreten. Die Fluktuationen sind dabei in positive wie negative Richtung sichtbar. Um sie im Rauschsignal auswerten zu können, lässt sich die DSNU von Kamerasensoren aus den Dunkelbildern ermitteln. Dazu kann man aus mehreren Dunkelbildern den Durchschnittswert jedes Pixels berechnen, um die Abweichungen zwischen den Durchschnittswerten einzelner Pixel in den Dunkelbildern sichtbar zu machen. N entspricht dabei der Anzahl an Dunkelbildern und D dem Signalwert eines jeden Pixels. Es gilt:

$$\text{DSNU}_{\text{mean}} = \frac{1}{N} \sum_{k=1}^N D_k \quad (6.14)$$

Der Mean-Wert zeigt dabei die durchschnittlichen Rauschwerte eines jeden Pixels an, wodurch sich Schwankungen zwischen einzelnen Pixeln darstellen lassen.

Bei der Herstellung von Kamerasensoren kann es durch Fertigungsungenauigkeiten zu defekten Pixeln kommen. Hotpixel sind Bildpunkte, die durch einen Pixelfeh-

ler deutlich hellere Signalwerte ausgeben. Deadpixel sind Pixel, die deutlich niedrigere oder keine Signalwerte ausgeben. Um Hot- und Deadpixel zu ermitteln, können die Maximal- und Minimalwerte aller Pixel der Dunkelbilder miteinander verglichen werden. Hotpixel berechnen sich wie folgt:

$$DSNU_{\max} = \max_{k=1}^N D_k \quad (6.15)$$

Hotpixel werden dadurch sichtbar, dass ihre Signalwerte deutlich über den Werten der umliegenden Pixel liegen. Da der Übergang von überdurchschnittlich empfindlichen Pixeln zu defekten Hotpixeln oft fließend ist, wird bei der Analyse meist mit Schwellenwerten gearbeitet.

Deadpixel lassen sich entsprechend durch das Minimum der einzelnen Pixelwerte ermitteln. Es gilt:

$$DSNU_{\min} = \min_{k=1}^N D_k \quad (6.16)$$

Die Signalwerte von toten Pixeln sind deutlich geringer und liegen damit unter den umliegenden Pixelwerten. Auch hier wird bei der Analyse mit Schwellenwerten gearbeitet. Der Quellcode zu den Berechnungen findet sich im Anhang D.7.

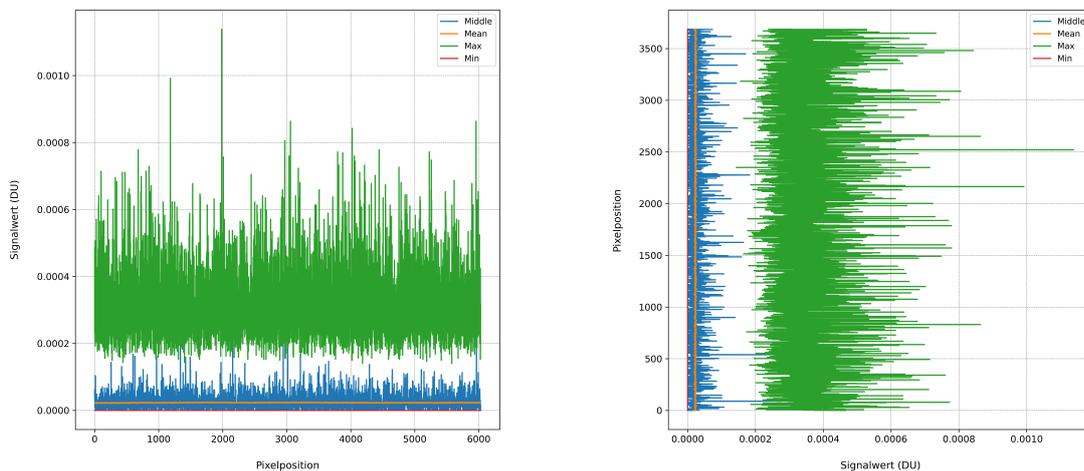


Abbildung 6.13.: DSNU der Sony A7IV

In Abbildung 6.13 ist das DSNU Profil der Sony A7IV dargestellt. In der linken Grafik werden alle DSNU-Werte spaltenweise zusammengefasst. In der rechten Grafik werden sie zeilenweise zusammengefasst. Der Maximalwert zeigt auf horizontaler und vertikaler Ebene einzelne überempfindliche Pixel, was für Hotpixel sprechen könnte. Ihr Signalwert ist jedoch nicht weit von den Schwankungen des Sen-

sors entfernt. Der Mittelwert der Pixel zeigt ein sehr einheitliches Bild ohne große Schwankungen in den Signalwerten. Die Standardabweichung in horizontaler Richtung beträgt $0,00000399\text{DU}$ und in vertikaler Richtung $0,00000417\text{DU}$. Die Standardabweichung ist ein Indikator für die Schwankung des Rauschsignals. Der durchschnittliche Mittelwert aller Pixel gibt das Blacklevel der Kamera an und beträgt bei der Sony A7IV $0,0000265\text{DU}$.

Abbildung C.22 zeigt das Profil der Sony A7III. Auch hier sind nur sehr leichte Unterschiede in der Empfindlichkeit einzelner Pixel zu sehen. Der Mittelwert der Pixel zeigt auch hier ein sehr einheitliches Bild ohne große Schwankungen in den Signalwerten. Die Standardabweichung in horizontaler Richtung beträgt $0,0000036\text{DU}$ und in vertikaler Richtung $0,0000041\text{DU}$. Der durchschnittliche Mittelwert aller Pixel beträgt bei der Sony A7III $0,0000366\text{DU}$.

Das DSNU-Profil der Panasonic Lumix GH5s zeigt einen etwas höheren durchschnittlichen Mittelwert als die Sony Kameras mit einem Signalwert von $0,0000673\text{DU}$, was auf ein höheres Blacklevel schließen lässt. Abbildung C.23 zeigt den Verlauf über die horizontal und vertikal verteilten Pixel. Die Schwankungen des Sensors in horizontaler Richtung betragen $0,00000846\text{DU}$. In vertikaler Richtung ergibt sich eine Standardabweichung von $0,00000868\text{DU}$.

In Abbildung C.24 ist der Verlauf der Phantom 4 Pro dargestellt. Bei der Phantom 4 Pro sind sehr deutliche Hotpixel zu sehen mit Signalwerten bis über $0,4\text{DU}$. Diese Hotpixel kommen durch defekte Pixel zustande, weshalb weiße Pixel in den Dunkelbildern zu sehen sind. Der durchschnittliche Mittelwert der Signalwerte liegt bei einem Blacklevel von $0,000287\text{DU}$. Die Schwankung des Sensors in horizontaler Richtung beträgt $0,000217\text{DU}$ und in vertikaler Richtung $0,000217\text{DU}$.

Die GoPro hat den höchsten durchschnittlichen Signalwerte aller fünf Kameras mit einem durchschnittlichen Signalwert von $0,000613\text{DU}$. Abbildung C.25 zeigt den Verlauf der GoPro 10. Die Schwankungen in horizontaler Richtung betragen $0,000439\text{DU}$ und in vertikaler Richtung $0,000437\text{DU}$.

6.2.6. Auswertung 10: PRNU

Die PRNU ist ein Indikator bei Kamerasystemen für die Abweichungen in den Signalwerten. Sie kann ermittelt werden, indem der Sensor gleichmäßig beleuchtet wird und die Abweichungen der Signalwerte aller Pixel von dem Mittelwert gemessen werden. Anschließend werden die Abweichungen in Untergruppen mit bestimmten Abständen eingeteilt und visuell dargestellt. Dadurch ergibt sich ein Überblick über die Abweichungen, die PRNU. Im Idealfall nutzt man für die Berechnung der PRNU eine Vielzahl von Signalbildern, um den Einfluss des Rauschens auf die Signalwerte zu minimieren und die Empfindlichkeit der einzelnen Pixel zu separieren. In der folgenden Auswertung wurde der PRNU von einem einzelnen Signalbild er-

mittelt und das durchschnittliche Rauschmuster der Dunkelbilder abgezogen. Auch die Vignettierung der Kamerasysteme nimmt Einfluss auf das Ergebnis in der PRNU Auswertung. Daher werden die äußeren 500 Pixel aus jeder Richtung abgeschnitten, um den Einfluss zu minimieren. Vollständig verhindern lässt sich der Effekt jedoch nicht (vgl. Auswertung 6.2.2). Der Pythoncode der Auswertung findet sich in Anhang D.8

Die Sony A7IV zeigt ein außergewöhnliches Signalmuster, da mehr Signalwerte bei einer Abweichung von -20% bis -40% als in positive Richtung von 20% bis 40% vorhanden sind. Dabei handelt es sich auf Grund der logarithmischen Darstellung jedoch um relativ wenige Werte und ist auf das Rauschmuster des Signals zurückzuführen. Die Standardabweichung zwischen den einzelnen Pixeln gibt an, wie stark die Streuung der Signalwerte ist. Sie liegt bei der A7IV bei einem Wert von 1,12%. Abbildung C.26 zeigt die PRNU der Sony A7IV.

Die Sony A7III zeigt ein sehr ähnliches Signalmuster wie die Sony A7IV. Sie ist um den Mittelwert sehr gleichabständig verteilt, mit einer Streuung des Signalwerts von 1,44%. Der Verlauf ist in Abbildung C.27 dargestellt.

Im Vergleich zu den Sony Kameras schneidet die GH5s sehr viel besser ab, was die Streuung um den Signalwert angeht. Das kann auch darauf zurückzuführen sein, dass die Vignettierung durch das Cropping des Lichtkegels des Objektivs sehr viel geringere Abweichungen verursacht. Mit einer Schwankung des Signalbereichs von 0,3% ist die Lumix GH5s die am geringsten schwankende Kamera im Test. Abbildung C.28 stellt die PRNU der Panasonic Lumix GH5s dar

Die Phantom 4 Pro und die GoPro 10 zeigen beide eine sehr starke Abweichung vom Mittelwert. Die Phantom 4 Pro hat eine Standardabweichung von 2,52% und ist in Abbildung C.29 dargestellt. Die GoPro 10 hat eine Standardabweichung von 3,68% und ist in Abbildung C.30 dargestellt.

6.2.7. Auswertung 11: Farbtreue

Um die Farbtreue von Kamerasystemen zu ermitteln, kann nicht wie in Auswertung 6.1.4 die Farbdifferenz ΔE von den Messwerten zum Referenzwert berechnet werden. Dafür gibt es folgende zwei Gründe:

Kameras mit Rohdatenzugriff geben die Rohdaten des Sensors aus, die in einem kameraspezifischen und nicht standardisierten Encoding vorliegen. Sie können nur durch aufwendige Kameracharakterisierungen und Farbraumtransformationen in den gleichen Farbraum transformiert werden. Diese Transformationsschritte der Hersteller sind oft nicht öffentlich zugänglich und würden daher den Rahmen dieser Auswertung sprengen. Abbildung 4.4 im Grundlagenkapitel 4 zeigt die Transformationsschritte, die dafür notwendig wären.

Bei den Broadcastkameras in diesem Test liegen die Sensordaten durch ihre Farbverarbeitung und die Transferkurven nicht in der szenenlinearen Domäne vor, da sie als HLG-Workflow in BT.2020 aufgezeichnet wurden. Da die Transformationschritte durch die Kameracharakterisierungen der einzelnen Kameras nicht eindeutig reproduzierbar sind, können keine direkten Farbwerte verglichen werden. Jedoch lassen sich durch die gemessenen XYZ-Werte des Spektrometers die Daten in der displaylinearen Domäne auswerten. Zwar lässt sich keine Aussage über die tatsächliche Farbverarbeitung des Sensors treffen, aber es lässt sich die Farbverarbeitung der Kamera mit einer simulierten HLG-BT.2020 Kamera vergleichen, die aus den Messdaten errechnet wurde. Außerdem lassen sich Farbshifts zwischen den Kamerasystemen vergleichen, die eine Aussage darüber erlauben, welche Farbbereiche beim Kameramatching besonders beachtet werden müssen.

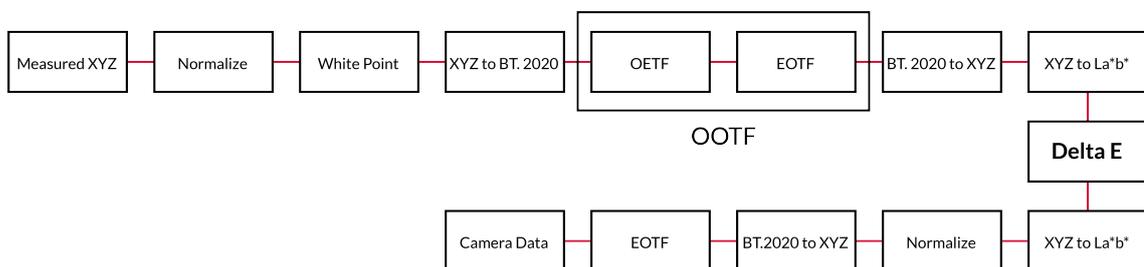


Abbildung 6.14.: Pipeline zum Vergleich von Kameradaten mit simulierten Referenzdaten in der displaylinearen Domäne

Abbildung 6.14 zeigt die Transformationsschritte, die bei den Referenzdaten des Spektrometers und den Kameradaten erbracht werden müssen, um die Daten miteinander zu vergleichen.

Um die Kameradaten und Messdaten zu vergleichen, müssen der Weißpunkt, der Farbraum und die Transferfunktion in einen vergleichbaren Zustand gebracht werden. Für den Vergleich werden die ΔE -Werte im La^*b^* -Farbraum bei einem Weißpunkt von D65 verglichen. Die Mess- und Kameradaten werden über die OOTF in HLG verglichen.

Die Kameradaten müssen dafür zunächst über die EOTF linearisiert werden. Anschließend wird der Farbraum und der Weißpunkt über die Matrizentransformation nach Bradford in das XYZ-Normfarbsystem bei einem Weißpunkt von D65 transformiert. Abschließend werden die Kameradaten in den La^*b^* -Farbraum transformiert, um dort die ΔE -Werte vergleichen zu können.

Die Messdaten benötigen einige zusätzliche Schritte. Da die Messdaten in linearer Abhängigkeit zu dem Licht der Szene stehen, kann von szenenlinearen Daten ausgegangen werden. Die gemessenen XYZ-Daten werden durch eine Weißpunkt- und Farbraumtransformation zunächst in den BT.2020 Farbraum transformiert. Dort kann über die HLG-OOTF von der szenenlinearen in die displaylineare Domäne transformiert werden. Anschließend werden die Daten über eine Rücktransformation von BT.2020 in XYZ und anschließend in La^*b^* transformiert.

Aus den ΔE -Werten lassen sich Farbverschiebungen zwischen der Panasonic UC4000, der Ikegami UHK-X700 und den simulierten HLG-Referenzkameradaten ermitteln. Die Grass Valley LDX135 wird in dieser Messung nicht berücksichtigt, da die Kamera im Testaufbau fälschlicherweise in einem HDR-SDR-Dual Workflow aufgebaut wurde und an dem Signalausgang daher ein Signal in BT.709 angelegt war.

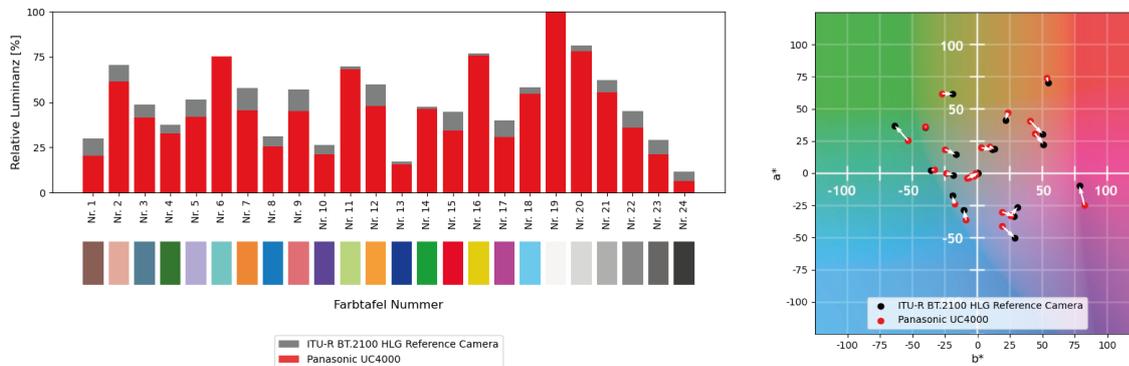


Abbildung 6.15.: Kameravergleich der Panasonic UC4000 mit den simulierten HLG-Kamerareferenzdaten

Vergleicht man die Panasonic UC4000 mit den transformierten Messdaten des Spektrometers, so ergeben sich die Helligkeitswerte und Farbverschiebungen, wie in Abbildung 6.15 dargestellt. Das Balkendiagramm links zeigt jeweils die relativen L-Werte zwischen der Kamera und den Messdaten. Die Darstellungen der Farbpatches zeigen nicht die exakten Farbwerte an und dienen nur zur Verdeutlichung. Es ist zu sehen, dass es durch den Systemlook der Kamera zu leichten Helligkeitsverschiebungen in der Kamera kommt. Fast alle Farbpatches fallen dabei dunkler aus als die simulierte Referenz. Bei den Farbverschiebungen ist zu sehen, dass vor allem die orangen, lila und pinken Farbtöne stärkere Abweichungen zeigen. Gegenüber der Referenz zeigen Patch 5, 7, 9 und 12 Farbverschiebungen von über $10 \Delta E$. In den entsättigten Farbtönen sowie den Grün- und Blautönen zeigt die Kamera eine gute Farbgenauigkeit gegenüber der Referenz. Patch 6, 8, 10, 11, 13, 14 und 18 und 24 zeigen einen Wert zwischen $3,8$ und $4,9 \Delta E$.

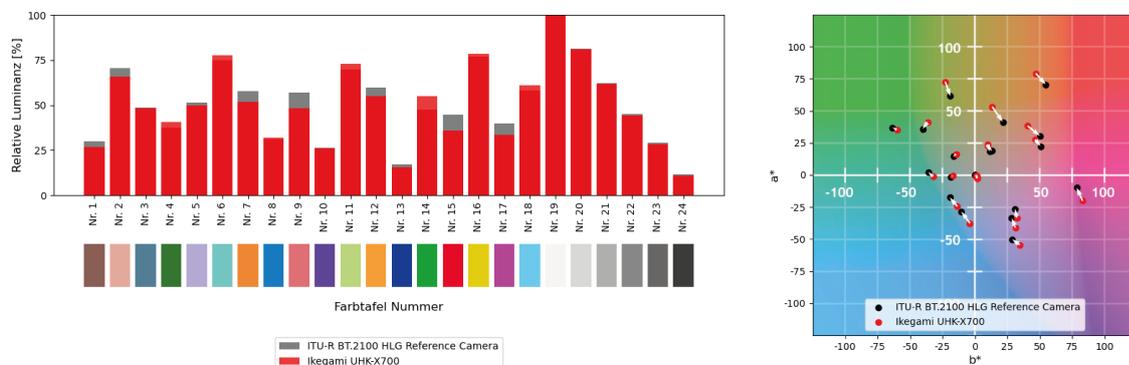


Abbildung 6.16.: Kameravergleich der Ikegami UHK-X700 mit den simulierten HLG-Kamerareferenzdaten

Die Ikegami UHK-X700 zeigt gegenüber den simulierten Referenzdaten eine geringere Abweichung der Helligkeitswerte. Am stärksten ist die Abweichung vor allem in den Gelb-, Orange- und Rottönen. Die Blau- und Lilatöne zeigen nur kleine Differenzen in der Helligkeit. Auch in den Farbverschiebungen zeigen sich die größten Abweichungen in den Gelb, Orange und Rottönen. Patch 7, 9 und 15 zeigen die stärkste Abweichung mit 9-11 ΔE . Die kleinste Verschiebung zeigt sich in den bläulichen und grünlichen Farbtönen. Patch 1, 3, 4, 5, 8, 10, 11 und 13 liegen unter einem Wert von 5 ΔE , wobei Patch 3, 8 und 13 nur knapp über der Wahrnehmungsgrenze von 1 ΔE liegen.

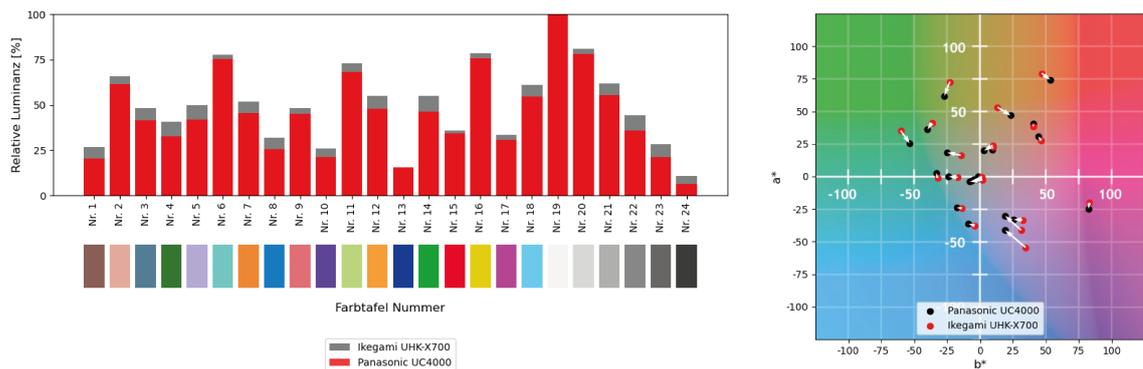


Abbildung 6.17.: Kameravergleich der Ikegami UHK-X700 mit den simulierten HLG-Kamerareferenzdaten

Vergleicht man die beiden Kameras direkt, zeigen sich Color Shifts und Helligkeitsverschiebungen zwischen den beiden Kamerasystemen. Abbildung 6.17 zeigt das Verhältnis der Messwerte zueinander. Wie schon bei dem Vergleich mit den simulierten HLG-Messdaten zeigt die Panasonic UC4000 auch gegenüber der Ikegami UHK-X700 dunklere Luminanzwerte bei den Farbpatches. In den Lila- und Grüntönen sieht man relativ starke Farbverschiebungen zwischen den beiden Kameras. Die Patches 3, 4, 5, 9, 12 und 14 haben dabei die stärksten Abweichungen bei einem ΔE zwischen 7 und 11. Die Farbpatches 2, 5, 13 und 16 zeigen die niedrigsten Farbverschiebungen und liegen vor allem in den Blau und Orangetönen. Sie haben ein ΔE zwischen 3 und 4.

6.3. Interpretation der Daten

Auf Grundlage der Anforderungen durch das HVS 2, durch den HDR-WCG Workflow 3 und aus den Messergebnissen in diesem Kapitel lässt sich die Forschungsfrage: „Welche Kameras aus dem aktuellen Produktionsumfeld erfüllen die Anforderungen an die visuelle Wahrnehmung des Menschen und die Anforderungen für HDR-WCG-Produktionen?“ wie folgt beantworten. Anforderungen eins bis vier definieren dabei die Anforderungen durch das HVS. Anforderungen fünf bis neun definieren die Anforderungen durch die SDR und HDR Standards.

6.3.1. Anforderung 1: Wahrnehmungsgrenze des Steady State

Anforderung 1 an die Kamerasysteme im Bezug auf das HVS lautet: „Kamerasysteme müssen szenenlinear einen Blendenumfang von 12,3 Blendenstufen erreichen, um die Wahrnehmungsgrenze des Steady State des HVS zu erreichen.“ Abbildung 6.18 zeigt eine Übersicht, welche Kameras die Anforderungen erfüllen (grün), welche Kameras die Anforderungen nicht erfüllen (rot) und bei welchen Kameras die Anforderungen auf Grundlage dieser wissenschaftlichen Arbeit nicht beantwortet werden kann (grau). Es wird bei der Auswertung von dem errechneten maximalen SNR bei Vollaussteuerung des Kamerasensors ausgegangen.

Sony A7III		Phantom 4 Pro			Grass Valley LDX135		Panasonic UC4000	
ISO 125	ISO 100	X	X	X				
Sony A7IV	Panasonic Lumix GH5s			GoPro 10				Ikegami UHK-X700

Abbildung 6.18.: Übersicht über Kamerasysteme, welche die Wahrnehmungsgrenze des Steady State erreichen können

Für die Sony A7IV und Sony A7III konnte in dieser Arbeit gezeigt werden, dass bei idealer Aussteuerung des Sensors die Wahrnehmungsgrenzen des HVS überschritten werden konnte. Die Sony A7IV erreicht einen Dynamikumfang von 13,43 Blendenstufen (80,55dB) bei einem ISO von 50. Die Sony A7III kann bei gleichem ISO 12,96 Blendenstufen (77,76dB) abdecken. Die Panasonic Lumix GH5s verfehlt die Wahrnehmungsgrenze knapp mit einem Dynamikumfang von 12,11 Blendenstufen (72,63dB). Bei der Phantom 4 Pro und der GoPro 10 ist das Ergebnis etwas deutlicher mit einem Dynamikumfang von 10,82 Blendenstufen (64,93dB) und 11,24 Blendenstufen (67,45dB).

Bei der Sony A7IV kann die Wahrnehmungsgrenze bis zu einer Vorverstärkung von ISO 125 überschritten werden. Bei ISO 125 wird die Wahrnehmungsgrenze erstmals unterschritten mit einem Dynamikumfang von 12,25 Blendenstufen (73,49dB). Die Sony A7III unterschreitet die Wahrnehmungsgrenze bei ISO 100 mit einem Dynamikumfang von 11,99 Blendenstufen (71,93dB).

6.3.2. Anforderung 2: Wahrnehmungsgrenze des Rauschens

Anforderung 2 beruht auf der vereinfachten Annahme, dass Rauschen ab einem Signal-Rauschabstand von 42dB wahrnehmbar wird. Sie lautet: „Kamerasysteme müssen einen Signal-Rauschabstand von 42dB vorweisen, um für das visuelle System die ungewichtete Wahrnehmungsgrenze des Rauschens zu erreichen.“ Abbildung 6.19 zeigt eine Übersicht, welche Kameras die Anforderungen bis zu welcher Vorverstärkung erfüllen. Auch hier wird von dem errechneten maximalen SNR bei Vollaussteuerung des Kamerasensors ausgegangen.

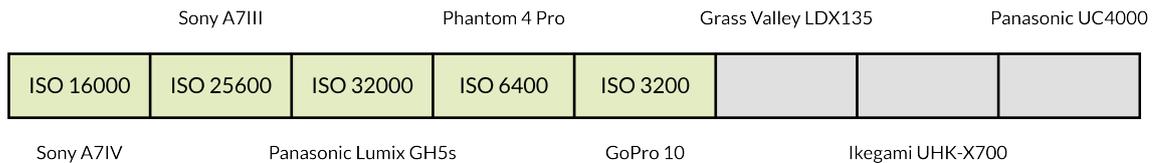


Abbildung 6.19.: Übersicht über Kamerasysteme die die Wahrnehmungsgrenze des Rauschens erreichen können

In dieser Arbeit konnte gezeigt werden ab welcher Vorverstärkung die Kameras ein sichtbares Rauschen erzeugen. Für die Sony A7IV ist der Punkt bei einer Vorverstärkung von ISO 16000 mit einem Dynamikumfang von 6,86 Blendenstufen (41,14dB) erreicht. Die Sony A7III hat eine etwas bessere Low-Light-Performance. Das Rauschen wird ab einer Vorverstärkung von ISO 25600 sichtbar mit einem Dynamikumfang von 6,80 Blendenstufen (40,79dB). Bei errechneter Vollaussteuerung hat die Panasonic Lumix GH5s die beste Low-Light-Performance gezeigt. Das Rauschen wird ab einer Vorverstärkung von ISO 32000 mit einem Dynamikumfang von 6,65 Blendenstufen (39,88dB) sichtbar. Die Phantom 4 Pro erreicht eine Vorverstärkung ISO 6400 für die Wahrnehmungsgrenze des Rauschens bei einem Dynamikumfang von 5,92 Blendenstufen (35,52dB). Bei der GoPro wird das Rauschen ab einer Vorverstärkung von ISO 3200 sichtbar bei einem Dynamikumfang von 6,86 Blendenstufen (41,17dB).

6.3.3. Anforderung 3: Farbdifferenzen zum Referenzfarbraum

Anforderung 3 ergibt sich aus den Ausführungen zum La^*b^* Farbraum und lässt sich auf Grund der Sichtbarkeitsgrenze bei Farbdifferenzen definieren: „Kamerasysteme müssen eine Farbdifferenz von kleiner $1 \Delta E$ im La^*b^* Farbraum zu den Referenzmessdaten vorweisen, damit keine visuellen Unterschiede sichtbar sind.“ In dieser Arbeit konnte gezeigt werden, dass Farbverschiebungen der Panasonic UC4000 und der Ikegami UHK-X700 für alle Farbpatches oberhalb der Sichtbarkeitsgrenze von $1 \Delta E$ liegen. Damit sind die Farbverschiebungen gemessen an einer simulierten HLG-Referenzkamera sichtbar. Die Ikegami UHK-X700 war dabei für die meisten Farbpatches (ausgenommen Patch 6, 11 und 14) in den Helligkeitswerten näher an den simulierten HLG-Referenzdaten. Bei den Farbverschiebungen zeigen sich deutliche Unterschiede zwischen den Kameras. Die Ikegami UHK-X700 neigt vor allem in den Gelb-, Orange- und Rottönen zu Farbverschiebungen in Richtung wärmerer Farben und in den Blautönen in Richtung der Grüntöne. Die Panasonic UC4000 zeigt vor allem Verschiebungen in den Rot-, Lila und Blautönen. Sie zeigen vorwiegend in Richtung der Lilatöne.

Auch wenn die Farbverschiebungen gegenüber der Referenzdaten gezeigt werden konnten, lässt sich aus dieser Arbeit keine Aussage darüber treffen, welchen Einfluss die technischen Komponenten der Kamerasysteme haben. Auch wenn alle Kamerasysteme auf ihre Werkseinstellungen zurückgesetzt wurden und Para-

meter, die einstellbar die Farbverschiebungen beeinflussen, abgeschaltet wurden, können die Farben durch den Systemlook der Kameracharakterisierungen beeinflusst werden. Die Auswertung ist als Interpretation des Farblooks der Hersteller gegenüber einer Referenz-HLG-Kamera zu verstehen.

6.3.4. Anforderung 4: Farbdifferenzen der Kamerasysteme

Anforderung 4 geht wie Anforderung 3 aus der Sichtbarkeitsgrenze von Farbdifferenzen im La^*b^* -Farbraum hervor und lautet: „Kamerasysteme müssen untereinander eine Farbdifferenz von kleiner $1 \Delta E$ im La^*b^* Farbraum vorweisen, damit keine visuellen Unterschiede zwischen den Kamerasystemen sichtbar sind.“ Auch bei den Farbverschiebungen konnte gezeigt werden, dass in allen Farbpatches leichte Farbverschiebungen visuell sichtbar sind. Die stärkste Verschiebung zwischen der Ikegami UHK-X700 und der Panasonic UC4000 findet sich in den Grün- und Lilatönen. Die Rot und Blautöne haben eine sehr gute gemeinsame Farabbildung. Um die Kameras aufeinander anzugleichen, müssten die Lilatöne der Ikegami UHK-X700 leicht entsättigt werden. Die Grüntöne der UC4000 zeigen eine Verschiebung in kalte Farbtöne. Bei den Gelbtönen gibt es zudem eine leichte Verschiebung in Richtung der warmen Farbtöne.

Die Farbtemperatur der Kamerasysteme nimmt Einfluss auf die Farabbildung der Farbpatches. Die Verschiebungen der Farben sind daher nicht unmittelbar auf jede Belichtungssituation übertragbar.

6.3.5. Anforderung 5: SDR-Workflow

Anforderung 5 ergibt sich aus den Ausführungen zum SDR-Workflow und lautet: „Kamerasysteme müssen szenenlinear einen Dynamikumfang von 6,74 Blendenstufen abbilden können, um für einen SDR-Workflow geeignet zu sein.“ Abbildung 6.20 zeigt eine Übersicht, welche Kameras die Anforderungen bis zu welcher Vorverstärkung erfüllen. Es wird von dem errechneten maximalen SNR bei Vollaussteuerung des Kamerasensors ausgegangen.

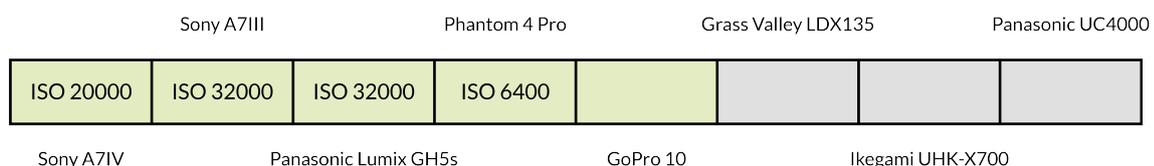


Abbildung 6.20.: Übersicht über Kamerasysteme, die einen SDR-Workflow abbilden können

In dieser Arbeit konnte gezeigt werden, dass grundsätzlich alle Consumerkameras den SDR-Workflow abbilden können. Die Sony A7IV kann die Anforderung

bis zu einer Vorverstärkung von ISO 20000 bei einem Dynamikumfang von 6,52 Blendenstufen (39,16dB) erreichen. Die Sony A7III kommt sogar bis zu einer Vorverstärkung von ISO 32000 bei einem Dynamikumfang von 6,49 Blendenstufen (38,91dB). Bei der Panasonic Lumix GH5s wird der SDR-Workflow ebenfalls bis zu einer Vorverstärkung von ISO 32000 mit einem Dynamikumfang von 6,61 Blendenstufen (39,66dB) abgedeckt. Die Phantom 4 Pro kann den SDR-Workflow bis zu einer Vorverstärkung von ISO 6400 abdecken. Das entspricht einem Dynamikumfang von 5,92 Blendenstufen (35,52dB). Die GoPro kann über das gesamte Vorverstärkungsspektrum inklusive der maximalen Vorverstärkung von ISO 3200 den SDR-Workflow abbilden.

6.3.6. Anforderung 6: HLG-Workflow

Aus den Grundlagen zum HLG-Workflow lässt sich die Anforderung 6 an die Kamerasysteme ableiten: „Kamerasysteme müssen szenenlinear einen Dynamikumfang von 14,67 Blendenstufen abbilden können, um für einen HLG-Workflow geeignet zu sein.“ Von den Consumerkameras erreicht keine Kamera den Dynamikumfang der für einen HLG-Workflow notwendig wäre.

6.3.7. Anforderung 7-9: PQ-Workflow bei 1000, 4000 und 10000 cd/m² Spitzenleuchtdichte

Aus den Grundlagen lassen sich für den PQ-Workflow drei Anforderungen (7-9) ableiten. Sie lauten:

- „Kamerasysteme müssen szenenlinear einen Dynamikumfang von 11,02 Blendenstufen abbilden können, um für einen PQ-Workflow mit einer Spitzenleuchtdichte von 1000 cd/m² geeignet zu sein.“
- „Kamerasysteme müssen szenenlinear einen Dynamikumfang von 12,82 Blendenstufen abbilden können, um für einen PQ-Workflow mit einer Spitzenleuchtdichte von 4000 cd/m² geeignet zu sein.“
- „Kamerasysteme müssen szenenlinear einen Dynamikumfang von 14,02 Blendenstufen abbilden können, um für einen PQ-Workflow mit einer Spitzenleuchtdichte von 10000 cd/m² geeignet zu sein.“

Da es sich bei PQ um einen szenenreferenzierten Ansatz handelt, lässt sich für die Kamerasysteme ausrechnen, bis zu welcher Spitzenhelligkeit sie Signalwerte darstellen können. Sie lässt sich ermitteln, indem die Spitzenhelligkeit errechnet wird, die bei einem Schwarzwert von 0,005 cd/m² dem Abstand des Dynamikumfangs der Kamera entspricht. Für die Sony A7IV ist der Weißwert bei einem Signalwert

von 0,663DU und einer entsprechenden Spitzenleuchtdichte von 6367,57 cd/m² erreicht. Damit kann Sie den PQ1000 und PQ4000 Workflow abdecken. Die Sony A7III erreicht eine Spitzenleuchtdichte von 4466,21 cd/m² mit einem entsprechenden Signalwert von 0,48DU. Sie kann damit auch den PQ1000 und PQ4000 Workflow abdecken. Die Panasonic Lumix GH5s kommt auf eine Spitzenleuchtdichte von 2319,91 cd/m² mit einem Signalwert von 0,265DU, was für den PQ4000 Workflow nicht ausreicht. Der PQ1000 Workflow kann abgedeckt werden. Die Phantom 4 Pro kann weder den PQ1000 noch den PQ4000 Workflow abdecken. Sie erreicht eine Spitzenleuchtdichte von 859,90 cd/m² bei einem Signalwert von 0,109DU. Die GoPro 10 erreicht eine Spitzenleuchtdichte von 1191,66 cd/m² bei einem Signalwert von 0,146DU. Damit kann Sie den PQ1000 workflow abdecken.

Auch für den PQ-Workflow können die Anforderungen von den Kameras bis zu einer bestimmten Vorverstärkung erreicht werden. Die folgende Abbildung 6.21 stellt die Vorverstärkungsgrenzen da, bis zu denen die Anforderungen erfüllt werden.

	Sony A7III		Phantom 4 Pro		Grass Valley LDX135		Panasonic UC4000	
PQ 1000	ISO 1000	ISO 400	ISO 400	X	ISO 200			
PQ 4000	ISO 100	ISO 80	X	X	X			
PQ 10000	X	X	X	X	X			
	Sony A7IV		Panasonic Lumix GH5s		GoPro 10		Ikegami UHK-X700	

Abbildung 6.21.: Übersicht über Kamerasysteme, die einen PQ-Workflow abbilden können

7. Abschluss

In dieser Arbeit wurde in drei Grundlagenkapiteln gezeigt, welche Anforderungen durch das HVS und durch den HDR-WCG-Workflow an Kamerasysteme gestellt werden können und durch welche Einflussfaktoren Luminanz- und Spektralmessverfahren beeinflusst werden. Zudem konnte durch sieben Messaufbauten gezeigt werden, wie Kamerasysteme im Allgemeinen auf diese Anforderungen überprüft werden können und zu welchen Ergebnissen die Messungen für die acht Kamerasysteme kommen.

In den folgenden Kapiteln soll auf die Limitationen und den Ausblick der Forschung eingegangen und die dritte Forschungsfrage beantwortet werden.

7.1. Fazit

In den Messungen der Broadcast- und Consumerkameras sind verschiedene Auswertungen durchgeführt worden. Dabei ist für die Berechnung mancher Ergebnisse ein Rohdatenzugriff auf die Kamerasensoren und für manche Ergebnisse ein Referenzfarbraum zur Interpretation der Daten notwendig. Die folgende Abbildung 7.1 zeigt, welche Auswertungen für die unterschiedlichen Kamerasysteme im Rahmen dieser Arbeit möglich war.

Mit Blick auf die Übersicht der Daten lässt sich die Forschungsfrage, welche Kameras aus dem aktuellen Produktionsumfeld die Anforderungen an die visuelle Wahrnehmung des Menschen und die Anforderungen für HDR-WCG-Produktionen erfüllen wie folgt beantworten.

Luminanzmessverfahren benötigen den Zugriff auf die Sensordaten, um eine Aussage über den Signal-Noise-Ratio (SNR), das Rauschmuster oder verschiedene Störeinflüsse wie die DSNU, PRNU oder die Vignettierung zu treffen.

Für die fünf Consumergeräte konnte gezeigt werden, wie ein solches Messverfahren für die Verifikation von Kameras für die HDR-Produktionen und die menschliche Wahrnehmung aussehen kann. Insgesamt sind zwei Kameras (Sony A7IV und Sony A7III) für den PQ4000-Workflow geeignet. Vier Kameras (Sony A7IV, Sony A7III, Panasonic Lumix GH5s und GoPro 10) können den PQ1000-Workflow abdecken. Keine der Kameras ist sensorseitig für den HLG1000-Workflow geeignet. Auch in Bezug auf die Wahrnehmung konnten die Kameras verifiziert werden. Insgesamt

	Sony A7III		Phantom 4 Pro		Grass Valley LDX135		Panasonic UC4000	
Wahrnehmung Helligkeit	ISO 125	ISO 100	X	X	X			
Wahrnehmung Rauschen	ISO 16000	ISO 25600	ISO 32000	ISO 6400	ISO 3200			
Delta E Referenz								
Delta E Kamera								
SDR	ISO 20000	ISO 32000	ISO 32000	ISO 6400				
HLG	X	X	X	X	X			
PQ 1000	ISO 1000	ISO 400	ISO 400	X	ISO 200			
PQ 4000	ISO 100	ISO 80	X	X	X			
PQ 10000	X	X	X	X	X			
Visual Noise								
	Sony A7IV	Panasonic Lumix GH5s		GoPro 10		Ikegami UHK-X700		

Abbildung 7.1.: Übersicht über die Ergebnisse der Interpretation der Daten

konnten zwei Kameras (Sony A7IV und Sony A7III) den Steady State der menschlichen Dynamikwahrnehmung voll ausschöpfen. Die Rauschgrenze wurde bei allen Kameras gut abgedeckt.

Spektralmessverfahren müssen in einem Referenzfarbraum durchgeführt werden, da natürliche Farben in verschiedenen Referenzfarbräumen unterschiedliche Farbpositionen haben und somit unterschiedlich wirken können. Daher können Kameras nur verglichen werden, wenn sie die Sensordaten in einem vergleichbaren Farbraum aufgezeichnet haben.

Für die Panasonic UC4000 und die Ikegami UHK-X700 konnten die Kameralooks in Bezug auf eine simulierte Referenz-Kamera im HLG BT.2020 Raum verglichen und eine Aussage über die Farbdifferenzen zwischen den Kameras getroffen werden. Zwar waren die Farbtonabweichungen für beide Kameras zur Referenz und auch untereinander im sichtbaren Bereich, jedoch konnte gezeigt werden, wie weit die Standard-Farblocks des Herstellers von den Referenzdaten abweichen. Aus der Praxis heraus können so Farbunterschiede sichtbar gemacht werden, die im Kameramatching korrigiert werden müssen.

7.2. Limitationen

Die Forschungsergebnisse zeigen einige Limitationen im Bezug auf die Datenerhebung und Auswertbarkeit der Daten.

In den Grundlagen wurden einige Annahmen getroffen, die komplizierte Wahr-

nehmungsaspekte vereinfachen. So wurde beispielsweise angenommen, dass Rauschen für das HVS erst ab einem SNR größer als 42dB sichtbar ist. Eine genauere Betrachtung wäre es, die Daten durch die vorgestellte Methode der Berechnung des Visual Noise zu erheben. Zudem wird für die HDR-Transferfunktionen davon ausgegangen, dass die Transferfunktionen unverändert als Ausgabemedium genutzt werden. Tatsächlich wird im Produktionskontext oft mit Anpassungen durch Highlightkompression, Erhöhung des Mittenkontrasts, etc. gearbeitet. Für solche Anwendungsfälle müssen Puffer in der Quantisierung der Daten bedacht werden, um Quantisierungsartefakte zu vermeiden.

Auch bei den Messungen gibt es verschiedene Einschränkungen, die Einfluss auf die Ergebnisse nehmen können. Bei der Ulbrichtkugel konnte eine leichte Helligkeits- und Farbverschiebung vor allem in den dunklen Belichtungsstärken gezeigt werden. Auch die Farbunterschiede zwischen den Referenzdaten des TE188 und der gemessenen Daten des Spektrometers zeigten Unterschiede im sichtbaren Bereich. Ein weiterer Aspekt ist, dass die Broadcastkameras direkt in einen Atomos Ninja aufgezeichnet wurden, wodurch es bei den Daten durch den 4:2:2 Output und der Kompression im ProResHQ Format zu Beeinflussungen der Farbdaten kommen kann. Die Grass Valley LDX 135 ist auf Grund einer fehlerhaften Aufzeichnung zudem vollständig aus den Messergebnissen herausgefallen.

In der Auswertung sind weitere Limitationen gezeigt worden. Alle Kameras mit Rohdatenzugriff zeigten teilweise starke Vignettierung, die Einfluss auf die Messwerte haben kann. Die Werte bewegten sich zwischen 6% bei der Panasonic Lumix GH5s bis hin zu 62% bei der GoPro10. Im Falle der Phantom 4 Pro konnten zudem Hotpixel gezeigt werden und leichte Empfindlichkeitsunterschiede der Pixel bei allen Kameras.

7.3. Ausblick

Abbildung 7.1 zeigt neben den messbaren Ergebnissen auch gut, welche Messwerte nicht erhoben werden konnten. Um die spektralen Eigenschaften aus den Sensordaten vollständig auswerten zu können, benötigt man ein Verfahren, das eine einheitliche Kameracharakterisierung für den Vergleich von Kameradaten mit Rohdatenzugriff ermöglicht. Dadurch könnten die Farbunterschiede, die durch einzelne Sensoren entstehen, besser mit einer Referenz verglichen werden.

Der Visual Noise Wert ist ein interessanter Maßstab, um das Rauschsignal auf die Sichtbarkeit in bestimmten Umgebungssituationen auszuwerten. Sie sind auf Grundlage der erhobenen Daten auswertbar und könnten die Genauigkeit der Deutung im Bezug auf die Rauschwahrnehmung erheblich verbessern.

Um die Broadcastkameras auf Grundlage ihrer Luminanzeigenschaften zu analysieren, ist ein Rohdatenzugriff notwendig. Eine Zusammenarbeit mit den Kamera-

herstellern könnte Sensorunterschiede und Unterschiede in der Verarbeitung der Daten durch die einzelnen Hersteller sichtbar machen. Eine Kooperation ist auf Grundlage der Betriebsgeheimnisse der Kamerahersteller zwar unrealistisch, könnte aber enorm dazu beitragen, Kameraunterschiede einfacher im Produktionsumfeld auszugleichen.

Abkürzungsverzeichnis

HDR High Dynamic Range

WCG Wide Color Gamut

SDR Standard Dynamic Range

SCG Standard Color Gamut

PQ Perceptual Quantizer

HLG Hybrid Log Gamma

HVS Human Visual System

CCU Camera Control Unit

RCP Remote Control Panel

PoE Power over Ethernet

fL Footlambert

SNR Signal-Noise-Ratio

DSNU Dark Signal Non-Uniformity

PRNU Photo-Response Non-Uniformity

DU Digital Units

dB Dezibel

CIE Commission internationale de l'Éclairage

CCD Charge Coupled Device

CMOS Complementary Metall Oxid Semiconductor

MOS Metall Oxid Semiconductor

PPS Passive Pixel Sensor

RoI Region of Interest

CDS Correlated Double Sampling

CSF Contrast Sensitivity Function

PQ Perceptual Quantizer

HLG Hybrid Log Gamma

OETF Opto-Electronic Transfer Function

EOTF Electro-Optical Transfer Function

OOTF Opto-Optical Transfer Function

BBC British Broadcasting Company

NHK Nippon Hōsō Kyōkai

Arib Association of Radio Industries and Businesses

SMPTE Society of Motion Pictures and Television Engineers

Abbildungsverzeichnis

1.1. Bildgebungspipeline, frei nach [22, Abb. 1.1], [13, Abb 2.1], [57, Abb. 3] und [23, Abb. 1]	2
2.1. Übersicht zum Grundlagenkapitel Human Visual System	5
2.2. Reflexion, Remission und Brechung von Licht, frei nach [51, Abb. 2.4 und 2.6].	7
2.3. Albumcover des Albums „Dark Side of the Moon“ der Band Pink Floyd aus dem Jahre 1973 [2]	7
2.4. Absorption und Emission von Photonen, frei nach [51, Abb. 2.3] . . .	8
2.5. Grundgrößen der Fotometrie	9
2.6. Visualisierung des photometrisches Entfernungsgesetzes	10
2.7. Spektrale Eigenschaften verschiedener Lichtquellen (natürliches Licht, Glühlampen, Quecksilberdampfampe und Mehrdampfampe) und eines reflektierenden Colorcharts, frei nach [35], [24]	11
2.8. Aufbau des menschlichen Auges und Ganglienzellen, frei nach [44] und [19]	12
2.9. Dunkeladaption des menschlichen Auges [25]	13
2.10. Wahrnehmbarer Helligkeitsbereich frei nach [20] und [51]	14
2.11. Transformationsschritte zur Herleitung des visuellen Rauschen [4] .	16
2.12. Spektrale Empfindlichkeit des menschlichen Auges [3]	17
3.1. Übersicht zum Grundlagenkapitel High Dynamic Range und Wide Color Gamut, frei nach [46]	19
3.2. Schematische Darstellung der Transferfunktionen der SDR, HLG und PQ Übertragungsstandards (freie Darstellung aus persönlichem Gespräch mit Justus Mai und nach [59] [61] [60] [58])	22
3.3. Das CIE-1931 Chromatizitätsdiagramm mit Farbräumen, frei nach [7], [1]	25
3.4. Farbraumtransformation über das CIE-Normfarbsystem	26
4.1. Ladungstransfer von CCD-Zellen [51, Abb. 6.13]	34
4.2. Grundlegender Aufbau einer CMOS-Zellen [51, Abb. 6.8]	35
4.3. Störeinflüsse beim Bildwandler und der Optik, frei nach [51, Abb. 6.31]	36
4.4. Bildpipeline von Kamerasystemen, frei nach [39]	39
4.5. Kameraoptiken von Broadcast- und Consumerkameras	40
4.6. Die Schärfentiefe von Kameraoptiken [8, S. 38]	41
5.1. Der Messaufbau im Dunkelraum der Lang AG	47
5.2. Blockschaltbild des Messaufbaus	48

5.3. Blockschaltbild der ersten und zweiten Messreihe	51
5.4. Blockschaltbild der dritten Messreihe	52
5.5. Blockschaltbild der fünften Messreihe	54
5.6. Blockschaltbild der ersten Messung aus der sechsten Messreihe . . .	55
5.7. Blockschaltbild der zweiten Messung aus der sechsten Messreihe . .	56
6.1. gemessene Leuchtdichte des Spektrometers gegenüber der emittierten Lichtstärke der Ulbrichtkugel	58
6.2. gemessene Leuchtdichte des Spektrometers gegenüber der emittierten Lichtstärke der Ulbrichtkugel	59
6.3. gemessene Farbtemperatur des Spektrometers gegenüber der emittierten Lichtstärke der Ulbrichtkugel	60
6.4. gemessene Farbtemperatur des Spektrometers gegenüber der emittierten Lichtstärke der Ulbrichtkugel	61
6.5. gemessene optische Dichte des TE223 im Vergleich zu den Herstellerangaben	62
6.6. Die gemessenen Farbpositionen des Farbpatches im Vergleich zur Referenz	64
6.7. Gammakurve und Sättigungskapazität der Sony A7IV	67
6.8. Gammakurve der UC4000	68
6.9. Beispielbild für Vignettierung und Berechnung der Entfernung einzelner Pixel	69
6.10. Vignette der Sony A7IV	70
6.11. Ausleserauschen der Sony A7IV	72
6.12. Signal-Rauschabstand aller Consumerkameras	76
6.13. DSNU der Sony A7IV	77
6.14. Pipeline zum Vergleich von Kameradaten mit simulierten Referenzdaten in der displaylinearen Domäne	80
6.15. Kameravergleich der Panasonic UC4000 mit den simulierten HLG-Kamerareferenzdaten	81
6.16. Kameravergleich der Ikegami UHK-X700 mit den simulierten HLG-Kamerareferenzdaten	81
6.17. Kameravergleich der Ikegami UHK-X700 mit den simulierten HLG-Kamerareferenzdaten	82
6.18. Übersicht über Kamerasysteme, welche die Wahrnehmungsgrenze des Steady State erreichen können	83
6.19. Übersicht über Kamerasysteme die die Wahrnehmungsgrenze des Rauschens erreichen können	84
6.20. Übersicht über Kamerasysteme, die einen SDR-Workflow abbilden können	85
6.21. Übersicht über Kamerasysteme, die einen PQ-Workflow abbilden können	87
7.1. Übersicht über die Ergebnisse der Interpretation der Daten	90
C.1. Gammakurve und Sättigungskapazität der Sony A7IV	XLI
C.2. Gammakurve und Sättigungskapazität der Sony A7III	XLI

C.3. Gammakurve und Sättigungskapazität der Lumix GH5s	XLII
C.4. Gammakurve und Sättigungskapazität der Phantom 4 Pro	XLII
C.5. Gammakurve und Sättigungskapazität der GoPro 10	XLIII
C.6. Vignette der Sony A7IV	XLIII
C.7. Vignette der Sony A7III	XLIV
C.8. Vignette der Lumix GH5s	XLIV
C.9. Vignette der Phantom 4 Pro	XLV
C.10. Vignette der GoPro 10	XLV
C.11. Ausleserauschen der Sony A7IV	XLVI
C.12. Ausleserauschen der Sony A7III	XLVI
C.13. Ausleserauschen der Lumix GH5s	XLVII
C.14. Ausleserauschen der Phantom 4 Pro	XLVII
C.15. Ausleserauschen der GoPro 10	XLVIII
C.16. Signal-Rauschabstand der Sony A7IV	XLVIII
C.17. Signal-Rauschabstand der Sony A7III	XLIX
C.18. Signal-Rauschabstand der Lumix GH5s	XLIX
C.19. Signal-Rauschabstand der Phantom 4 Pro	L
C.20. Signal-Rauschabstand der GoPro 10	L
C.21. DSNU der Sony A7IV	LI
C.22. DSNU der Sony A7III	LI
C.23. DSNU der Panasonic Lumix GH5s	LII
C.24. DSNU der Phantom 4 Pro	LII
C.25. DSNU der GoPro 10	LIII
C.26. PRNU der Sony A7IV	LIII
C.27. PRNU der Sony A7III	LIV
C.28. PRNU der Lumix GH5s	LIV
C.29. PRNU der Phantom 4 Pro	LV
C.30. PRNU der GoPro 10	LV

Tabellenverzeichnis

3.1. Bradford-Matrix	28
5.1. Gemessene Kamera-Objektiv-Kombinationen	47
6.1. Gemessene Farbpositionen der Farbpatches und des Weißpatches des TE188	63
6.2. gemessene Sensorkapazität in cd/m^2	68
6.3. gemessene Signalwertverluste durch Vignettierung der Kameras . . .	71
6.4. SNR Daten der Consumerkameras	75
A.1. Graustufenwerte des TE223 nach den Herstellerangaben [17]	XVII
A.3. Spektraldaten der Farb- und Luminanzpatches des TE188 nach Her- stellerangaben [16]	XVIII
B.1. erste lineare Messung zur Linearitätsmessreihe der Ulbrichtkugel . .	XIX
B.2. zweite lineare Messung zur Linearitätsmessreihe der Ulbrichtkugel .	XX
B.3. dritte lineare Messung zur Linearitätsmessreihe der Ulbrichtkugel .	XXI
B.4. vierte lineare Messung zur Linearitätsmessreihe der Ulbrichtkugel .	XXII
B.5. erste logarithmische Messung zur Linearitätsmessreihe der Ulbricht- kugel	XXIII
B.6. zweite logarithmische Messung zur Linearitätsmessreihe der Ulbricht- kugel	XXIII
B.7. dritte logarithmische Messung zur Linearitätsmessreihe der Ulbricht- kugel	XXIV
B.8. vierte logarithmische Messung zur Linearitätsmessreihe der Ulbricht- kugel	XXIV
B.9. erste lineare Messung zur Farbtemperaturmessung der Ulbrichtkugel	XXV
B.10. zweite lineare Messung zur Farbtemperaturmessung der Ulbrichtkugel	XXVI
B.11. dritte lineare Messung zur Farbtemperaturmessung der Ulbrichtkugel	XXVII
B.12. vierte lineare Messung zur Farbtemperaturmessung der Ulbrichtkugel	XXVIII
B.13. erste logarithmische Messung zur Farbtemperaturmessung der Ul- brichtkugel	XXIX
B.14. zweite logarithmische Messung zur Linearitätsmessreihe der Ulbricht- kugel	XXIX
B.15. dritte logarithmische Messung zur Linearitätsmessreihe der Ulbricht- kugel	XXX
B.16. vierte logarithmische Messung zur Linearitätsmessreihe der Ulbricht- kugel	XXX
B.17. Messung der Beleuchtungsdichten der einzelnen Lumapatches des TE223	XXXI

B.18.Messung der Farbeigenschaften der einzelnen Farbpatches des TE188XXXII	
B.19.Messung der Farbeigenschaften der Lichtquelle	XXXII
B.20.Beleuchtungslichtmessung der Lumapatches bei verschiedenen Be- leuchtungsstärken	XXXIII
B.21.Beleuchtungslichtmessung und berechnete Beleuchtungslichtdichten der Lumapatches bei verschiedenen Belichtungsstärken	XXXIII
B.22.Messwerte der Sony A7IV bei den Luminanzmessungen	XXXIV
B.23.Kameraeinstellungen der Sony A7IV bei den Luminanzmessungen . .	XXXIV
B.24.Messwerte der Sony A7III bei den Luminanzmessungen	XXXV
B.25.Kameraeinstellungen der Sony A7III bei den Luminanzmessungen . .	XXXV
B.26.Messwerte der Panasonic Lumix GH5s bei den Luminanzmessungen	XXXVI
B.27.Kameraeinstellungen der Panasonic Lumix GH5s bei den Luminanz- messungen	XXXVI
B.28.Messwerte der Phantom 4 Pro bei den Luminanzmessungen	XXXVII
B.29.Kameraeinstellungen der Phantom 4 Pro bei den Luminanzmessungen	XXXVII
B.30.Messwerte der GoPro 10 bei den Luminanzmessungen	XXXVIII
B.31.Kameraeinstellungen der GoPro 10 bei den Luminanzmessungen . .	XXXVIII
B.32.Kameraeinstellungen der Spektralmessungen	XXXIX

Literatur

- [1] Adoniscik. „Foto auf Wikimedia Commons.“ Zugriff am 06. September 2024. (2024), Adresse: <https://commons.wikimedia.org/w/index.php?curid=3803195>.
- [2] „Albumcover Dark Side of the Moon auf Spotify.“ Zugriff am 13. September 2024. (2024), Adresse: <https://i.scdn.co/image/ab67616d0000b273d3709135d1005baa36939>
- [3] T. Anders. „Foto auf Wikimedia Commons.“ Zugriff am 13. September 2024. (2024), Adresse: <https://commons.wikimedia.org/w/index.php?curid=14869234>.
- [4] U. Artmann. „Noise: SNR vs. visual noise.“ Zugriff am 8. September 2024. (2013), Adresse: <https://www.image-engineering.de/library/technotes/706-noise-snr-vs-visual-noise>.
- [5] Association of Radio Industries and Businesses, *Parameter Values for the Hybrid Log-Gamma (HLG) High Dynamic Range Television (HDR-TV) System for Programme Production, Version 2.0*, Tokyo, Japan: Association of Radio Industries und Businesses, 2018. Adresse: https://www.arib.or.jp/english/std_tr/broadcasting/std-b67.html.
- [6] P. G. J. Barten, „Contrast Sensitivity of the Human Eye and its Effects on Image Quality,“ PhD Thesis, Institute for Perception Research, Eindhoven, 1999. DOI: 10.6100/IR523072.
- [7] BenQ. „Projector Color Coverage and Color Gamuts.“ Zugriff am 06. September 2024. (2024), Adresse: <https://www.benq.eu/de-de/knowledge-center/knowledge/projector-color-coverage-and-color-gamuts.html>.
- [8] P. Bühler, P. Schlaich und D. Sinner, *Digitale Fotografie, Fotografische Gestaltung, Optik, Kameratechnik* (Bibliothek der Mediengestaltung). Heidelberger Platz 3, 14197 Berlin, Germany: Springer Vieweg, 2017, S. x+95, ISBN: 978-3-662-53894-4. DOI: 10.1007/978-3-662-53895-1.
- [9] G. Cornuéjols, „Device for increasing the dynamic range of a camera,“ Pat. 5,638,119, 1991.
- [10] P. Corporation, *Operating Instructions - 4K Studio Camera, Model No. AK-UC4000G / AK-UC4000GS*, Osaka, Japan, 2018. Adresse: <https://pro-av.panasonic.net>.
- [11] Datacolor, *Color and Color Measurement: Book Three of Color Management - The CIE Color Space*. Datacolor, 2019.

- [12] D. T.-P. e.V. „Branchenkompass.“ Zugriff am 05. Juni 2024. (2024), Adresse: <https://tv-plattform.de/branchenkompass/>.
- [13] G. Eilertsen, „The high dynamic range imaging pipeline, Tone-mapping, distribution, and single-exposure reconstruction,“ PhD Thesis, Linköping University, 2018, ISBN: 978-91-7685-302-3.
- [14] I. Engineering. „Image Quality Factors.“ Zugriff am 6. September 2024. (2024), Adresse: <https://www.image-engineering.de/library/image-quality>.
- [15] I. Engineering, *Spherical Transparency Illuminator LE6-100*, LE6-100 Manual, Eitorf, Germany. Adresse: https://image-engineering.de/content/products/equipment/illumination_devices/le6/downloads/LE6-100_data_sheet.pdf.
- [16] I. Engineering, *TE188 Data Sheet*, TE188 A/D Manual, Eitorf, Germany. Adresse: https://www.image-engineering.de/content/products/charts/te188/downloads/TE188_V6_data_sheet.pdf.
- [17] I. Engineering, *TE223 Data Sheet*, TE223 D Manual, Eitorf, Germany. Adresse: https://www.image-engineering.de/content/products/charts/te223/downloads/TE223_D_data_sheet.pdf.
- [18] European Machine Vision Association, *EMVA Standard 1288: Standard for Characterization of Image Sensors and Cameras, Release 4.0 Linear*, Version 4.0 Linear, European Machine Vision Association, 16. Juni 2021. Adresse: <https://www.emva.org/>.
- [19] U. Eysel, *Physiologie des Menschen: mit Pathophysiologie*, R. Brandes, F. Lang und R. F. Schmidt, Hrsg. Springer Berlin Heidelberg, 2019, S. 1056, ISBN: 978-3-662-56468-4. DOI: 10.1007/978-3-662-56468-4_57.
- [20] J. A. Ferwerda, S. N. Pattanaik, P. Shirley und D. P. Greenberg, „A Model of Visual Adaptation for Realistic Image Synthesis,“ S. 10, 1996. DOI: 10.1145/237170.237263. Adresse: https://jamesferwerda.com/wp-content/uploads/2015/06/j03_ferwerda96_sig.pdf.
- [21] B. Fowler, M. D. Godfrey und S. Mims, „Reset Noise Reduction in Capacitive Sensors,“ *IEEE Transactions on Circuits and Systems I: Regular Papers*, Jg. 53, Nr. 8, S. 1658–1670, 2006. DOI: 10.1109/TCSI.2006.877890.
- [22] J. Fröhlich, „Encoding High Dynamic Range and Wide Color Gamut Imagery,“ PhD Thesis, Universität Stuttgart, 2017.
- [23] N. J. Gadgil, Q. Song, G.-M. Su und S. N. Hulyalkar, „The technology behind the exceptional visual experience via high dynamic range,“ *SIP*, Jg. 7, S. 10, 2018. DOI: 10.1017/ATSIP.2018.21. Adresse: <https://doi.org/10.1017/ATSIP.2018.21>.
- [24] Gigahertz-Optik. „Spectra of Various Light Sources.“ Zugriff am 13. September 2024. (2024), Adresse: <https://www.gigahertz-optik.com/en-us/service-and-support/knowledge-base/basics-light-measurement/light-color/light-sources-spectra/>.

- [25] E. GmbH. „Das menschliche Auge - Adaptation.“ Zugriff am 13. September 2024. (2023), Adresse: <https://www.ercoco.com/de/licht-planen/lichtwissen/das-menschliche-auge/adaptation-7486/>.
- [26] S. GmbH. „Produktsupport / FAQs - Kameras.“ Zugriff am 5. September 2024. (2024), Adresse: <https://www.sigma-foto.de/service/produktsupport-faqs/kameras/>.
- [27] M. Granados, B. Ajdin, M. Wand, C. Theobalt, H.-P. Seidel und H. P. A. Lensch, „Optimal HDR Reconstruction with Linear Digital Cameras,“ S. 215–222, 2010. DOI: 10.1109/CVPR.2010.5539863. Adresse: <https://doi.org/10.1109/CVPR.2010.5539863>.
- [28] D. J. Griffiths und D. F. Schroeter, *Introduction to Quantum Mechanics, Third Edition*, 3. Aufl. University Printing House, Cambridge CB2 8BS, United Kingdom: Cambridge University Press, 2018, S. 644, ISBN: 978-1-107-18963-8. DOI: 10.1017/978-1-316-99543-3.
- [29] Halbleiter.org. „Dotieren von Halbleitern.“ Zugriff am 05. September 2024. (2024), Adresse: <https://www.halbleiter.org/grundlagen/dotieren/#:~:text=Dotieren%20bedeutet%20das%20Einbringen%20von,Valenzelektronen%20%3D%205%2Dwertig>.
- [30] E. Hasche und P. Ingwer, *Game of Colors: Moderne Bewegtbildproduktion, Theorie und Praxis für Film, Video und Fernsehen*. Springer-Verlag Berlin Heidelberg, 2016, ISBN: 978-3-662-43888-6. DOI: 10.1007/978-3-662-43889-3.
- [31] K. H. Herrmann, *Der Photoeffekt, Grundlagen der Strahlenmessung*, 1. Aufl. Vieweg + Teubner Verlag Wiesbaden, 2013, S. VIII, 248, ISBN: 978-3-322-98787-7. DOI: 10.1007/978-3-322-98787-7.
- [32] C. Huygens, *Abhandlung über das Licht*. Leipzig, W. Engelmann, 1690, S. 114.
- [33] L. Ikegami Tsushinki Co., *UHK-X700 4K HDR Camera System Specification Sheet*, UHK-X700 Specification, Tokyo, Japan, 2023. Adresse: <https://www.ikegami.com>.
- [34] Imatest. „Image Quality Factors.“ Zugriff am 6. September 2024. (2024), Adresse: <https://www.imatest.com/docs/iqfactors/>.
- [35] K. Inoue, K. Hara und K. Urahama, „Spectral Reflectance Estimation and Color Reproduction Based on Sparse Neugebauer Model,“ *Advances in Science, Technology and Engineering Systems Journal*, Jg. 2, Nr. 3, S. 958–966, 2017. DOI: 10.25046/aj0203113. Adresse: <http://www.astesj.com/>.
- [36] International Commission on Illumination, „CIE 15:2004 Technical Report: Colorimetry, 3rd Edition,“ International Commission on Illumination, Vienna, Techn. Ber., 2004.
- [37] JADAK. „SpectraScan PR-670.“ Letzter Zugriff: 13.08.2024, 12:38 Uhr. (2024), Adresse: <https://www.jadaktech.com/product/spectrascan-pr-670/>.
- [38] P. R. a Jadak Brand, *PR-655/670 SpectraScan User Manual*, 7279 William Barry Blvd., North Syracuse, NY 13212, 2019. Adresse: <https://www.jadaktech.com/product/spectrascan-pr-670/>.

- [39] H. C. Karaimer und M. S. Brown. „A Software Platform for Manipulating the Camera Imaging Pipeline.“ Zugriff am 14. September 2024. (2016), Adresse: <https://karaimer.github.io/camera-pipeline/>.
- [40] T. Kunkel und E. Reinhard, „A Reassessment of the Simultaneous Dynamic Range of the Human Visual System,“ *ACM Transactions on Applied Perception*, Jg. 7, Nr. 3, S. 17–24, 2010. DOI: 10.1145/1836248.1836251. Adresse: <https://dl.acm.org/doi/10.1145/1836248.1836251>.
- [41] B. Lindbloom. „Bruce Lindbloom - Color Science Resources.“ Zugriff am 11. September 2024. (2024), Adresse: <http://www.brucelindbloom.com/>.
- [42] I. LLC. „Dokumentation Dynamic Range.“ Zugriff am 30. August 2024. (2024), Adresse: <https://www.imatest.com/support/docs/23-1/dynamic/>.
- [43] I. LLC. „ISO 15739 – Noise.“ Zugriff am 8. September 2024. (2024), Adresse: <https://www.imatest.com/solutions/iso-15739/>.
- [44] E. Lübbe, *Farbempfindung, Farbbeschreibung und Farbmessung, Eine Formel für die Farbsättigung*. Heidelberger Platz 3, 14197 Berlin, Germany: Springer Vieweg Berlin, Heidelberg, 2013, S. 213, ISBN: 978-3-8348-2228-4. DOI: 10.1007/978-3-8348-2228-4.
- [45] S. of Motion Picture und T. Engineers, „High Dynamic Range Electro-Optical Transfer Function of Mastering Reference Displays,“ SMPTE, SMPTE Standard SMPTE ST 2084:2014, 2014. DOI: 10.5594/SMPTE.ST2084.2014. Adresse: <https://ieeexplore.ieee.org/document/7291452>.
- [46] Neunzehn78. „Lichtabfall und das Abstandsgesetz.“ Zugriff am 13. September 2024. (2015), Adresse: <https://neunzehn78.info/2015/01/26/lichtabfall-und-das-abstandsgesetz/>.
- [47] J. Pade, *Quantenmechanik zu Fuß 2, Anwendungen und Erweiterungen*, 2. Aufl. Heidelberger Platz 3, 14197 Berlin, Germany: Springer Vieweg Berlin, Heidelberg, 2024, S. xx+643, ISBN: 978-3-662-67929-6.
- [48] T. Photometrics. „Pattern Noise: DSNU and PRNU.“ Zugriff am 6. September 2024. (2024), Adresse: <https://www.photometrics.com/learn/advanced-imaging/pattern-noise-dsnu-and-prnu>.
- [49] QUCam. „Correlated Double Sampling.“ Zugriff am 5. September 2024. (2024), Adresse: <http://www.qucam.com/cds.html>.
- [50] QUCam. „Reset Noise in CCD Cameras.“ Zugriff am 5. September 2024. (2024), Adresse: <http://www.qucam.com/resetnoise.html>.
- [51] U. Schmidt, *Professionelle Videotechnik: Grundlagen, Filmtechnik, Fernsehtechnik, Geräte- und Studiotechnik in SD, HD, UHD, HDR, IP*, 7. Aufl. Heidelberger Platz 3, 14197 Berlin, Germany: Springer Vieweg Berlin, Heidelberg, 2021, S. XIII, 914, ISBN: 978-3-662-63943-6. DOI: <https://doi.org/10.1007/978-3-662-63944-3>.
- [52] Society of Motion Picture and Television Engineers (SMPTE), *SMPTE RP 431-2:2011 - D-Cinema Quality – Reference Projector and Environment*, SMPTE Standard, 2011.

- [53] H. Song, H. Li und X. Liu, „Studies on Different Primaries for a Nearly-Ultimate Gamut in a Laser Display,“ *Optics Express*, S. 23 436–23 448, 2018. DOI: 10.1364/OE.26.023436.
- [54] I. O. for Standardization, *Photography – Electronic still picture imaging – Noise measurements*, Zugriff am 8. September 2024, 2023. Adresse: <https://www.iso.org/obp/ui/en/#iso:std:82233:en>.
- [55] S. S. Stevens, „The Psychological Review on the Psychophysical Law,“ *Jg. 64*, Nr. 3, S. 153–180, 1957.
- [56] P. A. Tipler und G. Mosca, *Physik, für Wissenschaftler und Ingenieure*, 7. Aufl. Heidelberger Platz 3, 14197 Berlin, Germany: Springer Spektrum Berlin, Heidelberg, 2015, S. xxxvi+1454, ISBN: 978-3-642-54165-0. DOI: <https://doi.org/10.1007/978-3-642-54166-7>.
- [57] I. T. Union, „Colour Conversation from Recommendation ITU-R BT.709 to Recommendation ITU-R BT.2020,“ ITU, ITU-R Recommendation ITU-R BT.2087-0, 2015. Adresse: <https://www.itu.int/rec/R-REC-BT.2087-0-201510-I/en>.
- [58] I. T. Union, „Image parameter values for high dynamic range television for use in production and international programme exchange,“ ITU, ITU-R Recommendation ITU-R BT.2100-2, 2018, Version 2. Adresse: <https://www.itu.int/rec/R-REC-BT.2100-2-201807-I/en>.
- [59] I. T. Union, „Parameter values for the HDTV standards for production and international programme exchange,“ ITU, ITU-R Recommendation ITU-R BT.709-6, 2015, Version 6. Adresse: <https://www.itu.int/rec/R-REC-BT.709-6-201506-I/en>.
- [60] I. T. Union, „Parameter values for ultra-high definition television systems for production and international programme exchange,“ ITU, ITU-R Recommendation ITU-R BT.2020-2, 2015, Version 2. Adresse: <https://www.itu.int/rec/R-REC-BT.2020-2-201510-I/en>.
- [61] I. T. Union, „Reference electro-optical transfer function for flat panel displays used in HDTV studio production,“ ITU, ITU-R Recommendation ITU-R BT.1886-0, 2011. Adresse: <https://www.itu.int/rec/R-REC-BT.1886-0-201103-I/en>.
- [62] G. Valley, *LDX 135/150 User Manual*, Montreal, Canada, 2023. Adresse: <https://www.grassvalley.com>.
- [63] Wisotop. „Lab: Farben vergleichen und Farbabstand.“ Zugriff am 11. September 2024. (2021), Adresse: <https://wisotop.de/farbabstand-farben-vergleichen.php>.
- [64] X. Yan, L. Yan und L. Guiling, „A Kind of Nonlinear Quantization Method to Extend the Color Gamut of DTV System,“ in *Proceedings of the 13th IEEE International Symposium on Consumer Electronics (ISCE 2009)*, Tianjin, China: IEEE, 2009, S. 141–143, ISBN: 978-1-4244-2976-9. DOI: 10.1109/ISCE.2009.5156953.

Ehrenwörtliche Erklärung

„Hiermit versichere ich, Jonas Christian Stolzmann ehrenwörtlich, dass ich die vorliegende Masterarbeit mit dem Titel: „Bewertung der Eignung verschiedener Broadcastkameras für den HDR-Live Workflow anhand aktueller Messmethodiken und Entwicklung von zwei Messverfahren für die Luminanz- und Spektralanalyse von Kamerasystemen“ selbstständig und ohne fremde Hilfe verfasst und keine anderen als die angegebenen Hilfsmittel benutzt habe. Die Stellen der Arbeit, die dem Wortlaut oder dem Sinn nach anderen Werken entnommen wurden, sind in jedem Fall unter Angabe der Quelle kenntlich gemacht. Die Arbeit ist weder vollständig noch in Teilen veröffentlicht oder in anderer Form als Prüfungsleistung vorgelegt worden.

Ich habe die Bedeutung der ehrenwörtlichen Versicherung und die prüfungsrechtlichen Folgen (§ 26 Abs. 2 Bachelor-SPO (6 Semester), § 24 Abs. 2 Bachelor-SPO (7 Semester), § 23 Abs. 2 Master-SPO (3 Semester) bzw. § 19 Abs. 2 Master-SPO (4 Semester und berufsbegleitend) der HdM) einer unrichtigen oder unvollständigen ehrenwörtlichen Versicherung zur Kenntnis genommen.“

Datum: Stuttgart, 16.09.2024 Unterschrift: J. Stolzmann

Gendererklärung

In dieser wissenschaftlichen Abschlussarbeit wird aufgrund der besseren Lesbarkeit bewusst auf eine geschlechtsneutrale Formulierung verzichtet. Sämtliche männliche Schreibweisen beziehen sich dabei gleichermaßen auf alle Geschlechter.

A. Anhang A: Herstellerangaben

A.1. Herstellerangaben des TE223

Die folgende Tabelle zeigt die Herstellerangaben für die Lumapatches des TE223.

Step	Density	Reflectance (%)
1	0.05	89
2	0.13	74
3	0.21	62
4	0.30	50
5	0.40	40
6	0.51	31
7	0.63	23
8	0.78	17
9	0.95	11
10	1.15	7
11	1.41	4
12	1.77	2
13	2.35	0.5

Tabelle A.1.: Graustufenwerte des TE223 nach den Herstellerangaben [17]

A.2. Herstellerangaben des TE188

Die folgende Tabelle zeigt die Herstellerangaben für die Farbpatches des TE188.

Patch	Name	CIE XYZ D50			CIE xy D50		CIE La*b* D50			Density	Transmission
A1	Dark skin	14.58	12.51	8.65	0.41	0.35	42.01	17.60	14.05	-	-
B1	Light skin	54.40	49.35	37.85	0.38	0.35	75.67	20.01	17.42	-	-
C1	Blue sky	15.34	17.61	30.46	0.24	0.28	49.02	-8.02	-18.71	-	-
D1	Foliage	7.14	12.41	3.20	0.31	0.55	41.85	-38.42	38.00	-	-
E1	Blue flower	49.31	45.51	85.69	0.27	0.25	73.23	17.16	-30.81	-	-
F1	Bluish green	41.71	57.74	67.46	0.25	0.35	80.59	-36.40	-3.96	-	-
A2	Orange	47.89	37.39	6.82	0.52	0.41	67.57	37.63	64.66	-	-
B2	Purplish blue	18.23	18.27	64.25	0.18	0.18	49.83	4.60	-54.26	-	-
C2	Moderate red	40.32	28.32	18.75	0.46	0.32	60.17	47.35	20.06	-	-
D2	Purple	13.53	8.25	38.88	0.22	0.14	34.49	43.46	-54.84	-	-
E2	Yellowgreen	60.19	82.73	33.24	0.34	0.47	92.89	-40.01	53.08	-	-
F2	Orange yellow	53.34	45.59	8.43	0.50	0.42	73.28	27.59	68.69	-	-
A3	Blue	8.60	3.75	45.32	0.15	0.07	22.84	57.04	-82.37	-	-
B3	Green	9.97	23.31	6.40	0.25	0.59	55.39	-71.87	45.33	-	-
C3	Red	37.11	19.08	2.55	0.63	0.32	50.78	77.58	57.92	-	-
D3	Yellow	55.78	63.14	5.06	0.45	0.51	83.51	-10.33	99.65	-	-
E3	Magenta	43.54	21.44	52.53	0.37	0.18	53.43	86.20	-37.16	-	-
F3	Cyan	47.09	59.26	102.19	0.23	0.28	81.43	-24.31	-27.83	-	-
A4	White	95.05	100.00	108.88	0.31	0.33	100.00	0.00	0.00	0.00	100.00
B4	Neutral	68.99	72.74	78.32	0.31	0.33	88.32	-0.33	0.67	0.19	64.57
C4	Neutral	41.54	43.83	46.90	0.31	0.33	72.12	-0.37	0.88	0.41	38.90
D4	Neutral	22.53	23.77	25.40	0.31	0.33	55.86	-0.28	0.78	0.68	20.89
E4	Neutral	11.07	11.68	12.37	0.32	0.33	40.70	-0.23	0.89	1.00	10.00
F4	Black	3.12	3.30	3.48	0.32	0.33	21.19	-0.21	0.65	1.52	3.02

Tabelle A.3.: Spektraldaten der Farb- und Luminanzpatches des TE188 nach Herstellerangaben [16]

B. Anhang B: Messungen

B.1. Messung 1

Messung 1.1 zeigt die Rohdaten der Linearitätsmessung des Leuchtmittels bei linearen Helligkeitsabständen. Insgesamt wurden vier Messungen durchgeführt die zeitlich voneinander getrennt waren. Die erste Messung fand am 1. Juli gegen 10 Uhr statt.

Belichtungsstärke Ulbrichtkugel (Lux)	Leuchtdichte Spektrometer (cd/m ²)
5170	2809
5000	2709
4800	2603
4600	2490
4400	2388
4200	2276
4000	2177
3800	2081
3600	1978
3400	1872
3200	1764
3000	1654
2800	1550
2600	1443
2400	1339
2200	1234
2000	1127
1800	1021
1600	908.2
1400	794.6
1200	682.6
1000	566.2
800	450.8
600	334.9
400	222.4
200	101.5

Tabelle B.1.: erste lineare Messung zur Linearitätsmessreihe der Ulbrichtkugel

Die zweite Messung fand am 1. Juli 2024 gegen 16 Uhr statt.

Belichtungsstärke Ulbrichtkugel (Lux)	Leuchtdichte Spektrometer (cd/m²)
5170	2811
5000	2710
4800	2603
4600	2489
4400	2388
4200	2277
4000	2178
3800	2081
3600	1978
3400	1873
3200	1763
3000	1653
2800	1551
2600	1442
2400	1338
2200	1233
2000	1128
1800	1021
1600	908.3
1400	795.7
1200	682.5
1000	567.4
800	450.5
600	335.2
400	222.4
200	101.7

Tabelle B.2.: zweite lineare Messung zur Linearitätsmessreihe der Ulbrichtkugel

Die dritte Messung fand am 2. Juli 2024 gegen 11 Uhr statt.

Belichtungsstärke Ulbrichtkugel (Lux)	Leuchtdichte Spektrometer (cd/m²)
5170	2796
5000	2711
4800	2600
4600	2491
4400	2382
4200	2272
4000	2176
3800	2079
3600	1975
3400	1870
3200	1762
3000	1655
2800	1548
2600	1442
2400	1336
2200	1231
2000	1127
1800	1019
1600	912.1
1400	798.3
1200	679.5
1000	569.2
800	452.8
600	336.7
400	222.7
200	103.3

Tabelle B.3.: dritte lineare Messung zur Linearitätsmessreihe der Ulbrichtkugel

Die vierte Messung fand am 2. Juli 2024 gegen 16:30 Uhr statt.

Belichtungsstärke Ulbrichtkugel (Lux)	Leuchtdichte Spektrometer (cd/m ²)
5170	2794
5000	2708
4800	2598
4600	2490
4400	2385
4200	2277
4000	2176
3800	2079
3600	1975
3400	1871
3200	1764
3000	1654
2800	1548
2600	1444
2400	1336
2200	1232
2000	1127
1800	1019
1600	909.3
1400	795.6
1200	681.7
1000	565.4
800	451.7
600	336.1
400	223.6
200	102

Tabelle B.4.: vierte lineare Messung zur Linearitätsmessreihe der Ulbrichtkugel

Messung 1.2 zeigt die Rohdaten der Linearitätsmessung des Leuchtmittels bei logarithmischen Helligkeitsabständen. Insgesamt wurden vier Messungen durchgeführt die zeitlich voneinander getrennt waren. Die erste Messung fand am 1. Juli gegen 11 Uhr statt.

Belichtungsstärke Ulbrichtkugel (Lux)	Leuchtdichte Spektrometer (cd/m ²)
5000	2706
2500	1388
1250	711.5
625	362
312	181.3
156	88.55
78	42.08
40	19.96
20	9.145

Tabelle B.5.: erste logarithmische Messung zur Linearitätsmessreihe der Ulbrichtkugel

Die zweite Messung fand am 1. Juli 2024 gegen 17:00 Uhr statt.

Belichtungsstärke Ulbrichtkugel (Lux)	Leuchtdichte Spektrometer (cd/m ²)
5000	2705
2500	1386
1250	719.8
625	361.5
312	181.1
156	88.73
78	41.51
40	19.93
20	9.047

Tabelle B.6.: zweite logarithmische Messung zur Linearitätsmessreihe der Ulbrichtkugel

Die dritte Messung fand am 2. Juli 2024 gegen 12:00 Uhr statt.

Belichtungsstärke Ulbrichtkugel (Lux)	Leuchtdichte Spektrometer (cd/m²)
5000	2705
2500	1386
1250	719.8
625	361.5
312	181.1
156	88.73
78	41.51
40	19.93
20	9.047

Tabelle B.7.: dritte logarithmische Messung zur Linearitätsmessreihe der Ulbrichtkugel

Die vierte Messung fand am 2. Juli 2024 gegen 17:30 Uhr statt.

Belichtungsstärke Ulbrichtkugel (Lux)	Leuchtdichte Spektrometer (cd/m²)
5000	2705
2500	1386
1250	719.8
625	361.5
312	181.1
156	88.73
78	41.51
40	19.93
20	9.047

Tabelle B.8.: vierte logarithmische Messung zur Linearitätsmessreihe der Ulbrichtkugel

B.2. Messung 2

Messung 2.1 zeigt die Rohdaten der Farbkonstanzmessung des Leuchtmittels bei linearen Helligkeitsabständen. Insgesamt wurden vier Messungen durchgeführt die zeitlich voneinander getrennt waren. Die erste Messung fand am 1. Juli gegen 10 Uhr statt.

Belichtungsstärke Ulbrichtkugel (Lux)	Farbtemperatur (K)
5170	3133
5000	3134
4800	3134
4600	3135
4400	3136
4200	3137
4000	3138
3800	3139
3600	3140
3400	3140
3200	3141
3000	3141
2800	3141
2600	3143
2400	3143
2200	3145
2000	3146
1800	3146
1600	3150
1400	3152
1200	3147
1000	3147
800	3148
600	3149
400	3152
200	3155

Tabelle B.9.: erste lineare Messung zur Farbtemperaturmessung der Ulbrichtkugel

Die zweite Messung fand am 1. Juli 2024 gegen 16:00 Uhr statt.

Belichtungsstärke Ulbrichtkugel (Lux)	Farbtemperatur (K)
5170	3133
5000	3134
4800	3135
4600	3135
4400	3135
4200	3138
4000	3139
3800	3139
3600	3139
3400	3140
3200	3141
3000	3141
2800	3141
2600	3143
2400	3144
2200	3144
2000	3145
1800	3147
1600	3149
1400	3152
1200	3147
1000	3148
800	3148
600	3149
400	3151
200	3154

Tabelle B.10.: zweite lineare Messung zur Farbtemperaturmessung der Ulbrichtkugel

Die dritte Messung fand am 2. Juli 2024 gegen 11:00 Uhr statt.

Belichtungsstärke Ulbrichtkugel (Lux)	Farbtemperatur (K)
5170	3133
5000	3134
4800	3134
4600	3135
4400	3135
4200	3137
4000	3138
3800	3139
3600	3139
3400	3140
3200	3140
3000	3141
2800	3141
2600	3143
2400	3144
2200	3145
2000	3146
1800	3147
1600	3150
1400	3152
1200	3148
1000	3148
800	3149
600	3151
400	3153
200	3155

Tabelle B.11.: dritte lineare Messung zur Farbtemperaturmessung der Ulbrichtkugel

Die vierte Messung fand am 2. Juli 2024 gegen 16:30 Uhr statt.

Belichtungsstärke Ulbrichtkugel (Lux)	Farbtemperatur (K)
5170	3133
5000	3134
4800	3135
4600	3135
4400	3136
4200	3137
4000	3138
3800	3139
3600	3139
3400	3140
3200	3141
3000	3140
2800	3141
2600	3143
2400	3144
2200	3145
2000	3146
1800	3147
1600	3150
1400	3153
1200	3147
1000	3147
800	3148
600	3150
400	3151
200	3155

Tabelle B.12.: vierte lineare Messung zur Farbtemperaturmessung der Ulbrichtkugel

Messung 2.2 zeigt die Rohdaten der Farbkonstanzmessung des Leuchtmittels bei logarithmischen Helligkeitsabständen. Insgesamt wurden vier Messungen durchgeführt die zeitlich voneinander getrennt waren. Die erste Messung fand am 1. Juli gegen 11 Uhr statt.

Belichtungsstärke Ulbrichtkugel (Lux)	Farbtemperatur (K)
5000	3134
2500	3143
1250	3153
625	3149
312	3154
156	3159
78	3157
40	3132
20	3105

Tabelle B.13.: erste logarithmische Messung zur Farbtemperaturmessung der Ulbrichtkugel

Die zweite Messung fand am 2. Juli 2024 gegen 17:00 Uhr statt.

Belichtungsstärke Ulbrichtkugel (Lux)	Farbtemperatur (K)
5000	3134
2500	3143
1250	3153
625	3149
312	3151
156	3157
78	3154
40	3129
20	3105

Tabelle B.14.: zweite logarithmische Messung zur Linearitätsmessreihe der Ulbrichtkugel

Die dritte Messung fand am 2. Juli 2024 gegen 12:00 Uhr statt.

Belichtungsstärke Ulbrichtkugel (Lux)	Farbtemperatur (K)
5000	3134
2500	3144
1250	3153
625	3151
312	3154
156	3158
78	3156
40	3130
20	3102

Tabelle B.15.: dritte logarithmische Messung zur Linearitätsmessreihe der Ulbrichtkugel

Die vierte Messung fand am 2. Juli 2024 gegen 17:30 Uhr statt.

Belichtungsstärke Ulbrichtkugel (Lux)	Farbtemperatur (K)
5000	3134
2500	3144
1250	3155
625	3150
312	3153
156	3161
78	3157
40	3132
20	3103

Tabelle B.16.: vierte logarithmische Messung zur Linearitätsmessreihe der Ulbrichtkugel

B.3. Messung 3

Jedes Lumapatch des Messcharts wurde bei maximaler Helligkeitsstufe was einer Belichtungsstärke von 6000 Lux des Leuchtmittels der Ulbrichtkugel entspricht vermessen. Dadurch ergeben sich die folgenden gemessenen Beleuchtungs-dichten.

Messchart	Messwert 1	Messwert 2
1	2045	6000
2	1801	6000
3	1510	6000
4	1253	6000
5	982.8	6000
6	769	6000
7	571.9	6000
8	405.1	6000
9	271.6	6000
10	173.2	6000
11	95.29	6000
12	38.71	6000
13	10.67	6000

Tabelle B.17.: Messung der Beleuchtungs-dichten der einzelnen Lumapatches des TE223

B.4. Messung 4

Das Messchart TE188 wurde bei einer Belichtungsstärke von 1500 Lux vermessen. Von jedem Farb- und Lumapatch wurden alle Parameter in der folgenden Tabelle zusammengefasst.

Patch Nummer	X	Y	Z	x	y	u'	v'	u	v	CCT
1	86,41	62,78	13,88	0,5299	0,385	0,3231	0,5282	0,3231	0,3521	1803
2	371,7	302	82,81	0,4913	0,3992	0,2887	0,5278	0,2887	0,2518	2233
3	133,4	136	85,92	0,3754	0,3828	0,2195	0,5035	0,2195	0,3356	4180
4	66,89	89,53	10,51	0,4007	0,5364	0,1856	0,559	0,1856	0,3727	4351
5	179,5	153,9	89,8	0,4242	0,3636	0,2605	0,5023	0,2605	0,3349	2867
6	264	320	153,2	0,3581	0,4341	0,1912	0,5214	0,1912	0,3476	4859
7	299,4	211,4	14,3	0,5702	0,4026	0,3409	0,5415	0,3409	0,361	1635
8	48,6	55,52	74,1	0,2727	0,3115	0,1761	0,4527	0,1761	0,3018	9619
9	308	203,5	39,76	0,5587	0,3692	0,354	0,5263	0,354	0,3509	NA
10	67,76	43,23	70,92	0,3725	0,2376	0,2918	0,4188	0,2918	0,2792	2429
11	264,8	291,9	52,29	0,4348	0,4793	0,2207	0,5473	0,2207	0,3649	3555
12	298,2	224,3	15,96	0,5538	0,4165	0,3215	0,544	0,3215	0,3627	1815
13	20,21	14,25	97,55	0,1531	0,108	0,1535	0,2436	0,1535	0,1624	NA
14	88,02	134	19,54	0,3643	0,5548	0,1632	0,5592	0,1632	0,3728	4982
15	281	137,8	3,138	0,666	0,3266	0,4768	0,5261	0,4768	0,3507	NA
16	389	365,4	13,67	0,5065	0,4757	0,2633	0,5563	0,2633	0,3709	2586
17	195,4	101,1	63,69	0,5426	0,2806	0,411	0,4781	0,411	0,3187	NA
18	158,6	189	116,4	0,3418	0,4074	0,1898	0,5089	0,1898	0,3392	5236
19	642,3	608,6	230,9	0,4335	0,4107	0,2455	0,5235	0,2455	0,349	3113
20	416,3	393,8	148,2	0,4344	0,4109	0,246	0,5237	0,246	0,3491	3099
21	240,5	227,4	85,38	0,4347	0,411	0,2462	0,5237	0,2462	0,3492	3094
22	128,7	121,6	45,5	0,4349	0,4112	0,2462	0,5239	0,2462	0,3492	3091
23	61,12	57,86	21,49	0,4351	0,4119	0,2461	0,5241	0,2461	0,3494	3093
24	17,75	16,8	6,169	0,4359	0,4126	0,2463	0,5245	0,2463	0,3497	3086

Tabelle B.18.: Messung der Farbeigenschaften der einzelnen Farbpatches des TE188

X	Y	Z	x	y	u'	v'	u	v	CCT
788,6	747,5	298,7	0,4298	0,4074	0,2446	0,5216	0,2446	0,3477	3150

Tabelle B.19.: Messung der Farbeigenschaften der Lichtquelle

B.5. Messung 5

Die Rohdaten der Gammakurvenmessungen wurden mit dem Spektrometer für jedes Lumachart bei allen Helligkeitsabstufungen erhoben. Die angegebenen Werte in den folgenden Tabellen für die Testcharts sind in cd/m^2 . Die erste Tabelle entspricht den gemessenen Werten, die zweite Tabelle entspricht den errechneten Werten.

Ulbrichtkugel (Lux)	Chart 1	Chart 2	Chart 3	Chart 4	Chart 5	Chart 6	Chart 7	Chart 8	Chart 9	Chart 10	Chart 11	Chart 12	Chart 13
5000	1654	1450	1226	1014	801,6	638,2	464	330,2	222,5	143,3	79,07	31,87	weak
2500	890,8	748	632,5	520,1	413,5	325,8	240,7	172,4	114,7	73,43	40,82	17,21	weak
1250	452,3	382	320,3	264	211,7	161,4	120,7	86,62	58,26	37,21	20,72	8,167	weak
625	222,6	186,6	156,4	129,1	102	78,92	59,21	42,39	28,31	18,36	10,16	weak	weak
312,5	109,6	92,1	78,24	63,91	50,62	38,94	28,88	20,84	13,92	8,98	weak	weak	weak
156,25	53,92	45,07	38,21	32,21	24,81	19,13	14,33	10,31	weak	weak	weak	weak	weak
78,125	26,5	22,19	18,82	15,38	12,23	9,756	weak	weak	weak	weak	weak	weak	weak
39,0625	12,7	10,79	9,12	7,512	weak	weak	weak	weak	weak	weak	weak	weak	weak
19,53125	6,363	weak	weak	weak	weak								

Tabelle B.20.: Beleuchtungsdichtemessung der Lumapatches bei verschiedenen Belichtungsstärken

Ulbrichtkugel (Lux)	Chart 1	Chart 2	Chart 3	Chart 4	Chart 5	Chart 6	Chart 7	Chart 8	Chart 9	Chart 10	Chart 11	Chart 12	Chart 13
5000	1654	1450	1226	1014	801,6	638,2	464	330,2	222,5	143,3	79,07	31,87	7,540833333
2500	890,8	748	632,5	520,1	413,5	325,8	240,7	172,4	114,7	73,43	40,82	17,21	3,770416667
1250	452,3	382	320,3	264	211,7	161,4	120,7	86,62	58,26	37,21	20,72	8,167	1,885208333
625	222,6	186,6	156,4	129,1	102	78,92	59,21	42,39	28,31	18,36	10,16	4,0835	0,942604167
312,5	109,6	92,1	78,24	63,91	50,62	38,94	28,88	20,84	13,92	8,98	5,08	2,04175	0,471302083
156,25	53,92	45,07	38,21	32,21	24,81	19,13	14,33	10,31	6,96	4,49	2,54	1,020875	0,235651042
78,125	26,5	22,19	18,82	15,38	12,23	9,756	7,165	5,155	3,48	2,245	1,27	0,5104375	0,117825521
39,0625	12,7	10,79	9,12	7,512	6,115	4,878	3,5825	2,5775	1,74	1,1225	0,635	0,25521875	0,05891276
19,53125	6,363	5,395	4,56	3,756	3,0575	2,439	1,79125	1,28875	0,87	0,56125	0,3175	0,127609375	0,02945638

Tabelle B.21.: Beleuchtungsdichtemessung und berechnete Beleuchtungsdichten der Lumapatches bei verschiedenen Belichtungsstärken

B.6. Messung 6

Die Rohdaten der Signalbilder und Dunkelbilder wurden für jede Kamera einzeln erhoben. Die Rohdaten der Messwerte und Kameraeinstellungen der einzelnen Kameras finden sich in den folgenden Tabellen.

Shutter	ISO	Spektrometer	Ulbrichtkugel
1/60	50	2721	5000
1/60	64	2142	3906,25
1/60	80	1729	3125
1/60	100	1396	2500
1/60	125	1132	2000
1/60	160	888,8	1562,5
1/60	200	715,4	1250
1/60	250	568,2	1000
1/60	320	452,7	781,25
1/60	400	362,9	625
1/60	500	291	500
1/60	640	227,3	390,625
1/60	800	181,8	312,5
1/60	1000	144,7	250
1/60	1250	114,6	200
1/60	1600	88,14	156,25
1/60	2000	69,85	125
1/60	2500	55	100
1/60	3200	42,21	78,125
1/60	4000	32,88	62,5
1/60	5000	25,57	50
1/60	6400	19,33	39,0625
1/60	8000	14,94	31,25
1/60	10000	11,86	25
1/60	12800	9,767	19,53125
1/80	16000	9,767	19,53125
1/100	20000	9,767	19,53125
1/125	25600	9,767	19,53125
1/160	32000	9,767	19,53125
1/200	40000	9,767	19,53125
1/250	51200	9,767	19,53125
1/320	64000	9,767	19,53125
1/400	80000	9,767	19,53125
1/500	102400	9,767	19,53125
1/640	128000	9,767	19,53125
1/800	160000	9,767	19,53125
1/1000	204800	9,767	19,53125

Tabelle B.22.: Messwerte der Sony A7IV bei den Luminanzmessungen

	Objektiv	Blende	Shutter	Brennweite	Schärfe	Adapter	Sensor	Abstand zur Lichtquelle
Sony GM 24-70 II	8.0	1/60	70	unendlich	nein	Full Frame	60	

Tabelle B.23.: Kameraeinstellungen der Sony A7IV bei den Luminanzmessungen

Shutter	ISO	Spektrometer	Ulbrichtkugel
1/60	50	2721	5000
1/60	64	2142	3906,25
1/60	80	1729	3125
1/60	100	1396	2500
1/60	125	1132	2000
1/60	160	888,8	1562,5
1/60	200	715,4	1250
1/60	250	568,2	1000
1/60	320	452,7	781,25
1/60	400	362,9	625
1/60	500	291	500
1/60	640	227,3	390,625
1/60	800	181,8	312,5
1/60	1000	144,7	250
1/60	1250	114,6	200
1/60	1600	88,14	156,25
1/60	2000	69,85	125
1/60	2500	55	100
1/60	3200	42,21	78,125
1/60	4000	32,88	62,5
1/60	5000	25,57	50
1/60	6400	19,33	39,0625
1/60	8000	14,94	31,25
1/60	10000	11,86	25
1/60	12800	9,767	19,53125
1/80	16000	9,767	19,53125
1/100	20000	9,767	19,53125
1/125	25600	9,767	19,53125
1/160	32000	9,767	19,53125
1/200	40000	9,767	19,53125
1/250	51200	9,767	19,53125
1/320	64000	9,767	19,53125
1/400	80000	9,767	19,53125
1/500	102400	9,767	19,53125
1/640	128000	9,767	19,53125
1/800	160000	9,767	19,53125
1/1000	204800	9,767	19,53125

Tabelle B.24.: Messwerte der Sony A7III bei den Luminanzmessungen

Objektiv	Blende	Shutter	Brennweite	Schärfe	Adapter	Sensor	Abstand zur Lichtquelle
Tamron 28-75	8.0	1/60	70	unendlich	nein	Full Frame	60

Tabelle B.25.: Kameraeinstellungen der Sony A7III bei den Luminanzmessungen

Shutter	ISO	Spektrometer	Ulbrichtkugel
1/60	80	1729	3125
1/60	100	1396	2500
1/60	125	1132	2000
1/60	160	888,8	1562,5
1/60	200	715,4	1250
1/60	250	568,2	1000
1/60	320	452,7	781,25
1/60	400	362,9	625
1/60	500	291	500
1/60	640	227,3	390,625
1/60	800	181,8	312,5
1/60	1000	144,7	250
1/60	1250	114,6	200
1/60	1600	88,14	156,25
1/60	2000	69,85	125
1/60	2500	55	100
1/60	3200	42,21	78,125
1/60	4000	32,88	62,5
1/60	5000	25,57	50
1/60	6400	19,33	39,0625
1/60	8000	14,94	31,25
1/60	10000	11,86	25
1/60	12800	9,767	19,53125
1/80	16000	9,767	19,53125
1/100	20000	9,767	19,53125
1/125	25600	9,767	19,53125
1/160	32000	9,767	19,53125
1/200	40000	9,767	19,53125
1/250	51200	9,767	19,53125
1/320	64000	9,767	19,53125
1/400	80000	9,767	19,53125
1/500	102400	9,767	19,53125
1/640	128000	9,767	19,53125
1/800	160000	9,767	19,53125
1/1000	204800	9,767	19,53125

Tabelle B.26.: Messwerte der Panasonic Lumix GH5s bei den Luminanzmessungen

Objektiv	Blende	Shutter	Brennweite	Schärfe	Adapter	Sensor	Abstand zur Lichtquelle
Sigma 18-35 1.8	8.0	1/60	35	Viltrox EF-M1 Adapter	nein	MFT	60

Tabelle B.27.: Kameraeinstellungen der Panasonic Lumix GH5s bei den Luminanzmessungen

Shutter	ISO	Spektrometer	Ulbrichtkugel
1/60	100	1396	2500
1/60	200	715,4	1250
1/60	400	362,9	625
1/60	800	181,8	312,5
1/60	1600	88,14	156,25
1/60	3200	42,21	78,125
1/60	6400	19,33	39,0625
1/60	12800	9,767	19,53125

Tabelle B.28.: Messwerte der Phantom 4 Pro bei den Luminanzmessungen

Blende	Shutter	Brennweite	Schärfe	Adapter	Sensor	Abstand zur Lichtquelle
8.0	1/60	weit		nein	1-Zoll	10

Tabelle B.29.: Kameraeinstellungen der Phantom 4 Pro bei den Luminanzmessungen

Shutter	ISO	Spektrometer	Ulbrichtkugel
1/60	100	1396	2500
1/60	200	715,4	1250
1/60	400	362,9	625
1/60	800	181,8	312,5
1/60	1600	88,14	156,25
1/60	3200	42,21	78,125

Tabelle B.30.: Messwerte der GoPro 10 bei den Luminanzmessungen

Blende	Shutter	Brennweite	Schärfe	Adapter	Sensor	Abstand zur Lichtquelle
	1/125	weit			6,17 x 4,55 mm	0

Tabelle B.31.: Kameraeinstellungen der GoPro 10 bei den Luminanzmessungen

B.7. Messung 7

Kamera	Type	Weißabgleich	Shutter	Blende	ISO	Brennweite	Lux Ulbrichtkugel
Sony A7IV	RAW	3100	60	F8	100	70	5000
Lumix GH5s	RAW	3100	60	F8	100	35	5000
Sony A7III	RAW	3100	60	F8	100	70	5000
Phantom 4 Pro	RAW	3100	60	F8	100	weit	5000
Go Pro	RAW	3100	125	F2.5	100	weit	5000
GV LDX135	HDR HLG	3100	60	F8	0	30	5000
Panasonic UC4000	HDR HLG	3100	60	F8	0	30	5000
Ikegami	HDR HLG	3100	60	F8	0	30	1250

Tabelle B.32.: Kameraeinstellungen der Spektralmessungen

C. Anhang C: Auswertungen

C.1. Auswertung 5

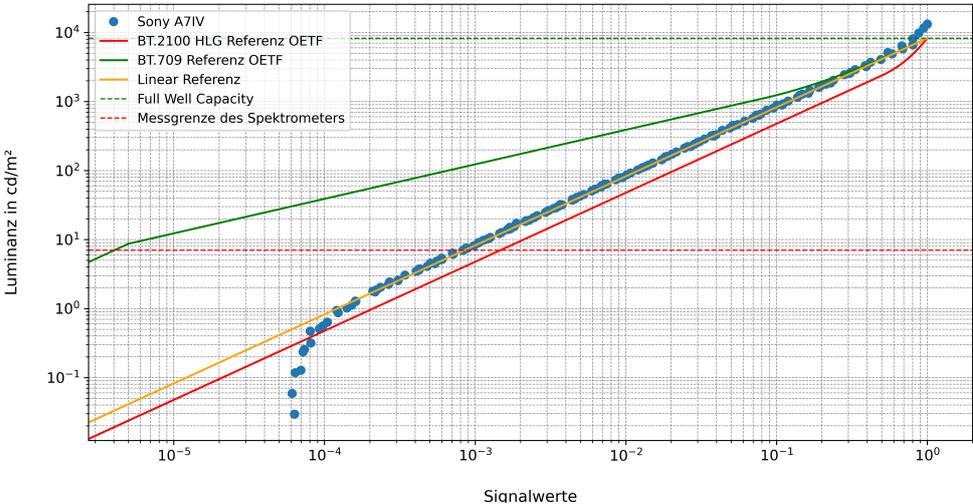


Abbildung C.1.: Gammakurve und Sättigungskapazität der Sony A7IV

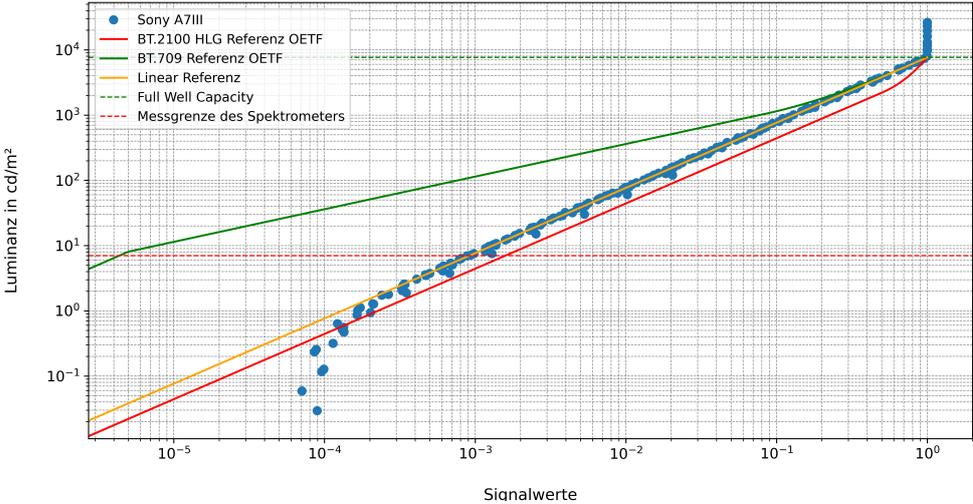


Abbildung C.2.: Gammakurve und Sättigungskapazität der Sony A7III

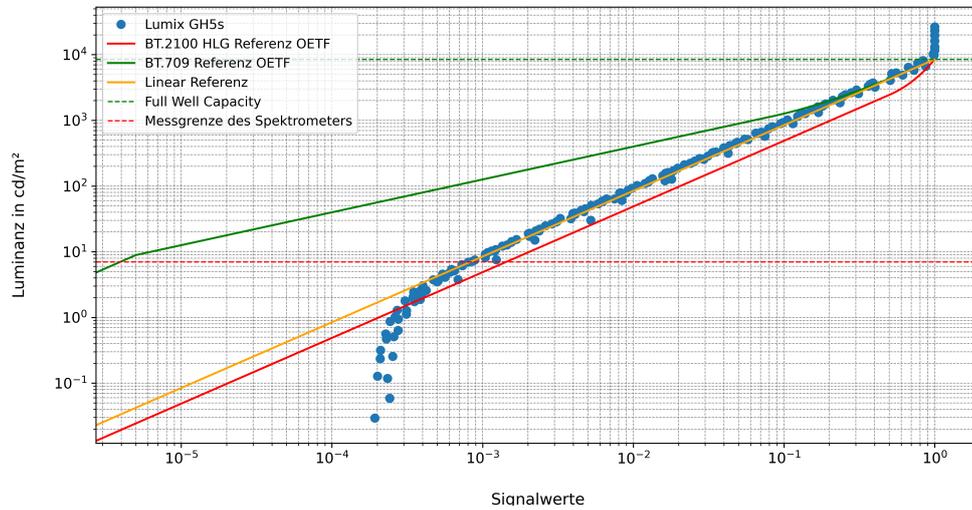


Abbildung C.3.: Gammakurve und Sättigungskapazität der Lumix GH5s

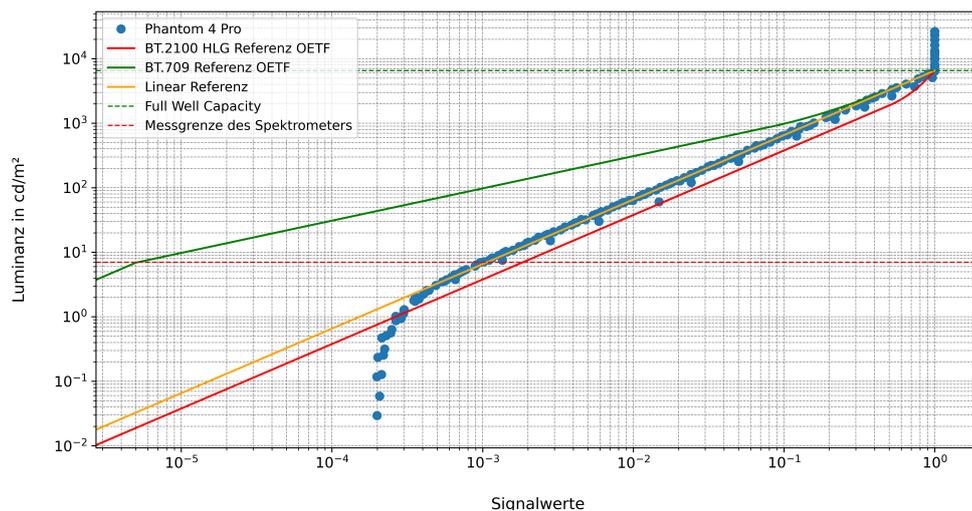


Abbildung C.4.: Gammakurve und Sättigungskapazität der Phantom 4 Pro

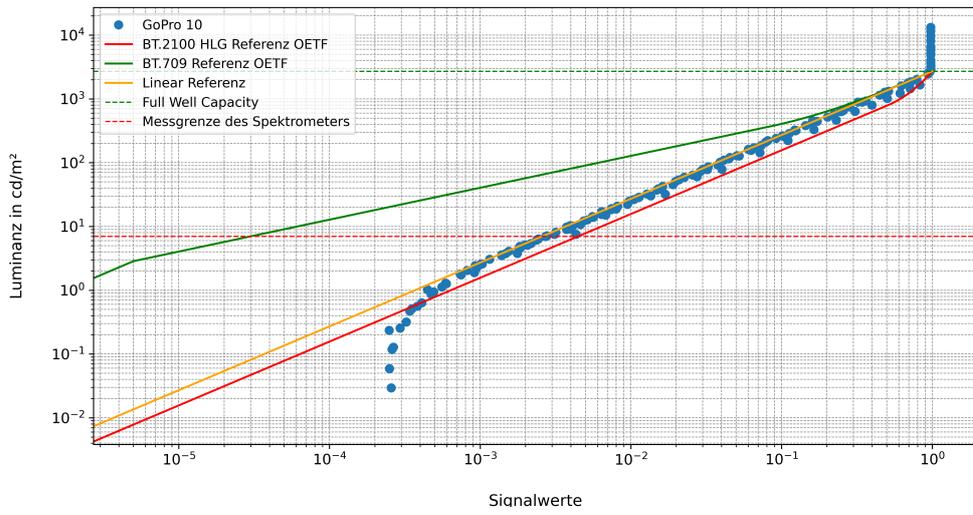


Abbildung C.5.: Gammakurve und Sättigungskapazität der GoPro 10

C.2. Auswertung 6

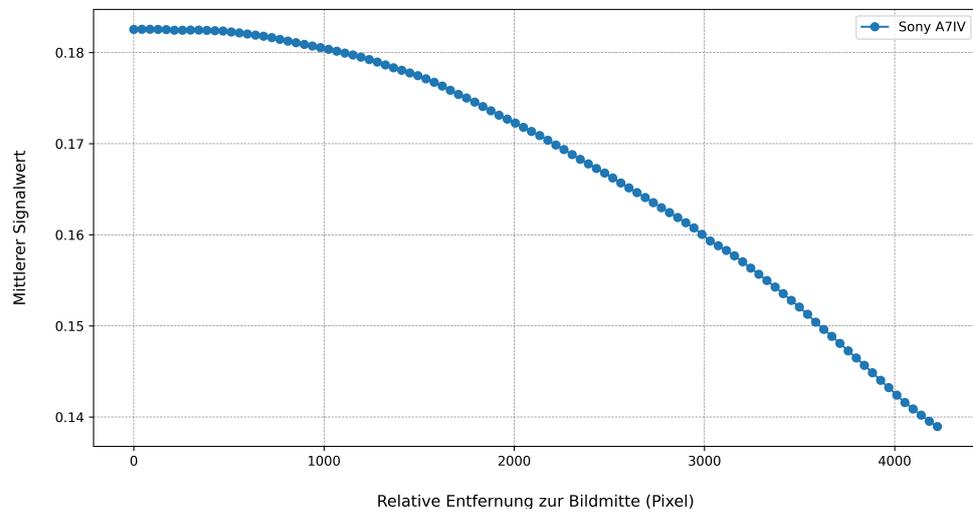


Abbildung C.6.: Vignette der Sony A7IV

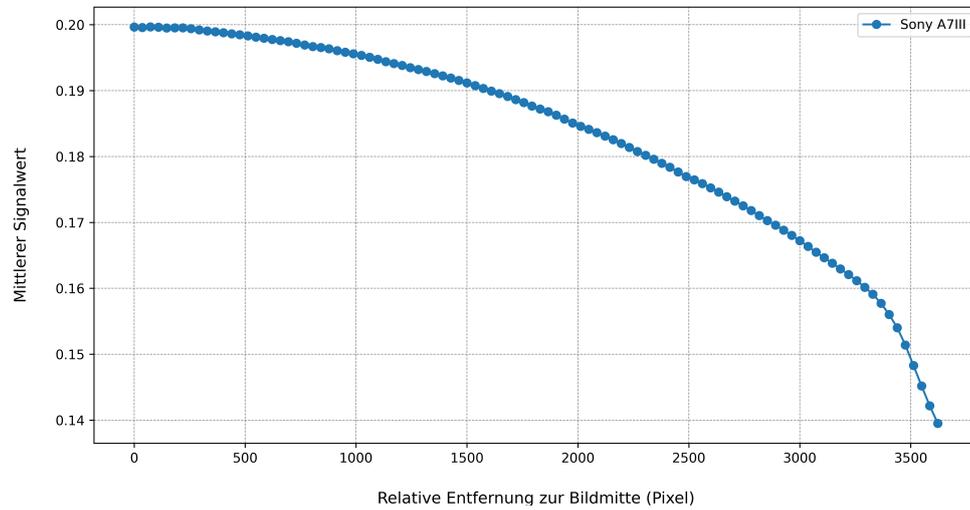


Abbildung C.7.: Vignette der Sony A7III

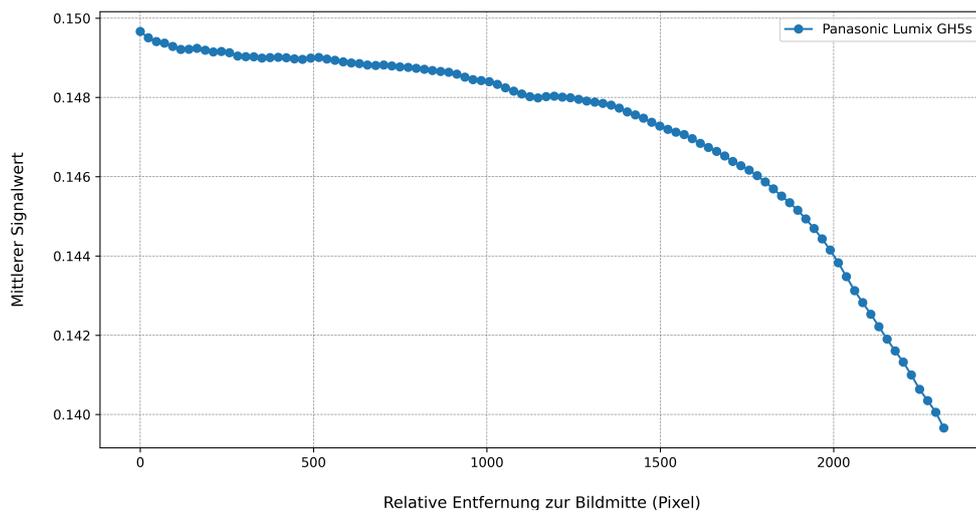


Abbildung C.8.: Vignette der Lumix GH5s

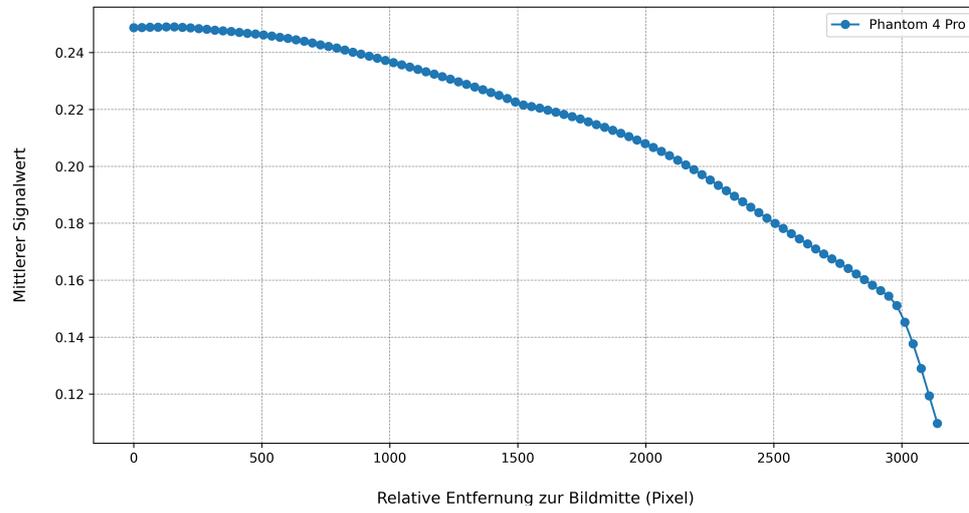


Abbildung C.9.: Vignette der Phantom 4 Pro

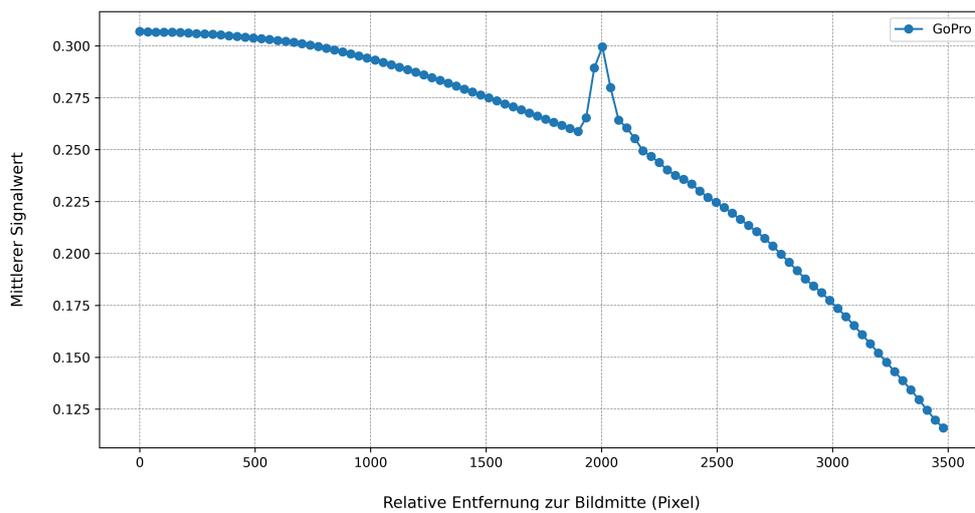


Abbildung C.10.: Vignette der GoPro 10

C.3. Auswertung 7

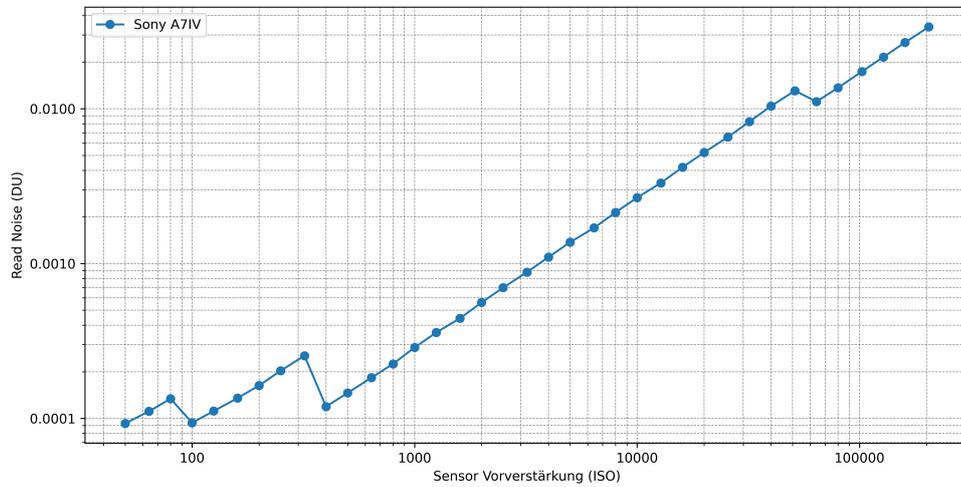


Abbildung C.11.: Ausleserauschen der Sony A7IV

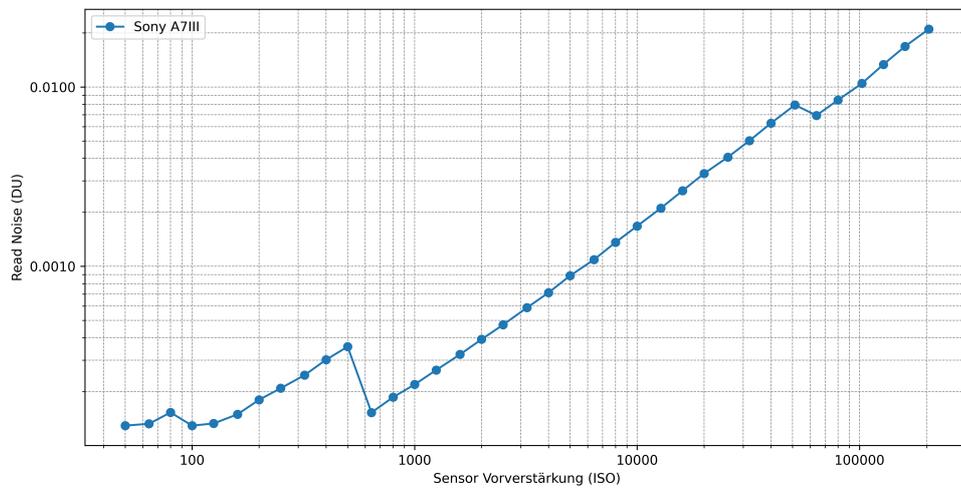


Abbildung C.12.: Ausleserauschen der Sony A7III

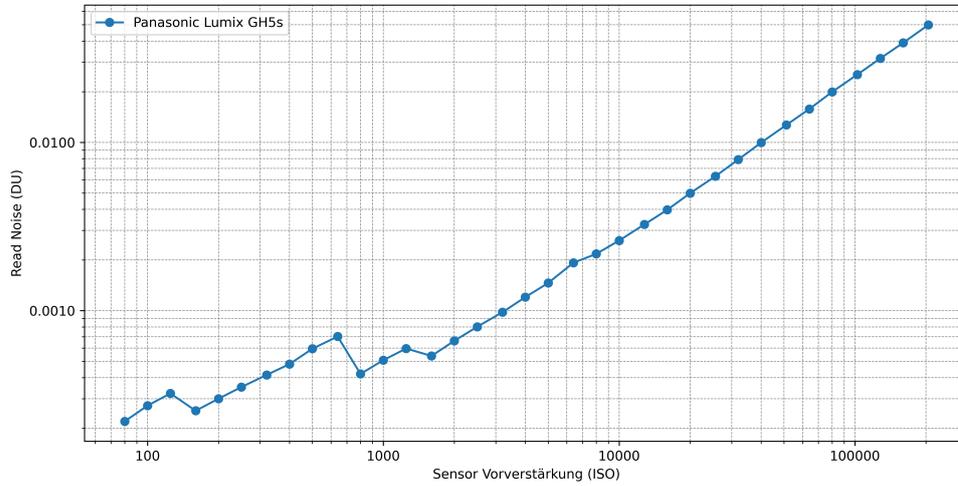


Abbildung C.13.: Ausleserauschen der Lumix GH5s

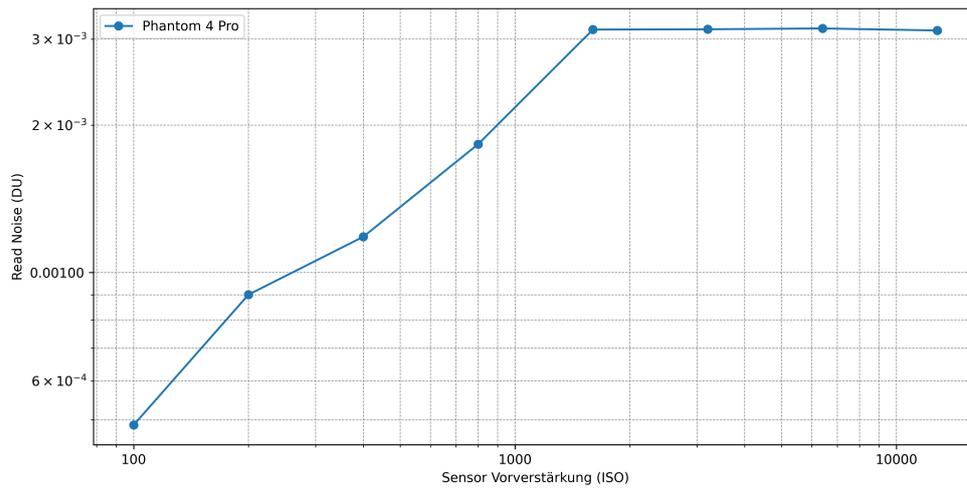


Abbildung C.14.: Ausleserauschen der Phantom 4 Pro

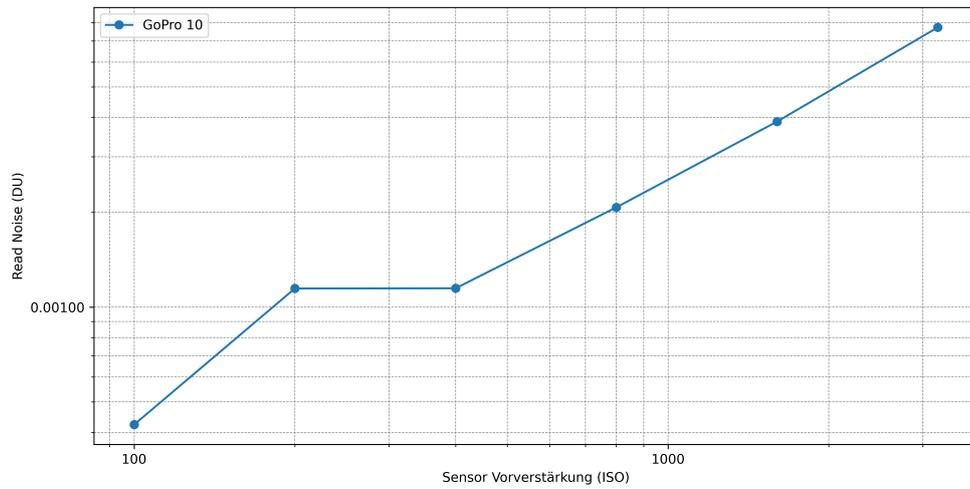


Abbildung C.15.: Ausleserauschen der GoPro 10

C.4. Auswertung 8

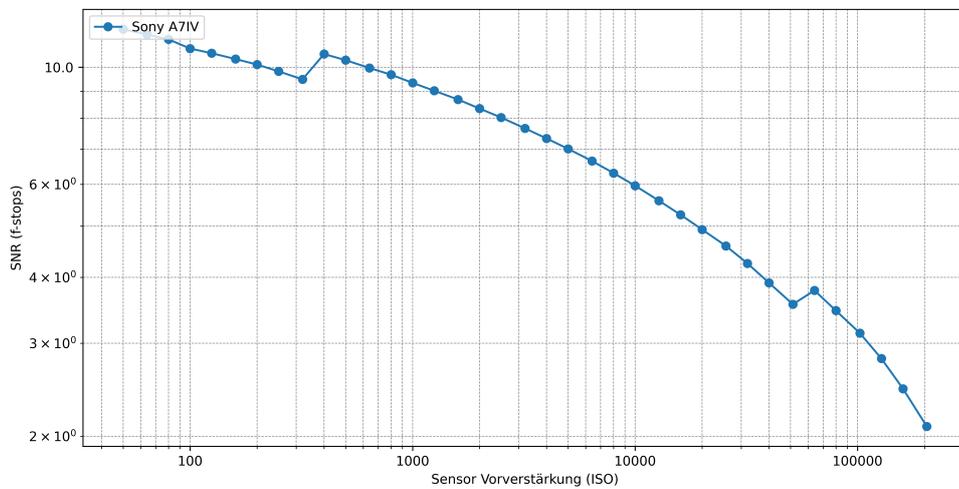


Abbildung C.16.: Signal-Rauschabstand der Sony A7IV

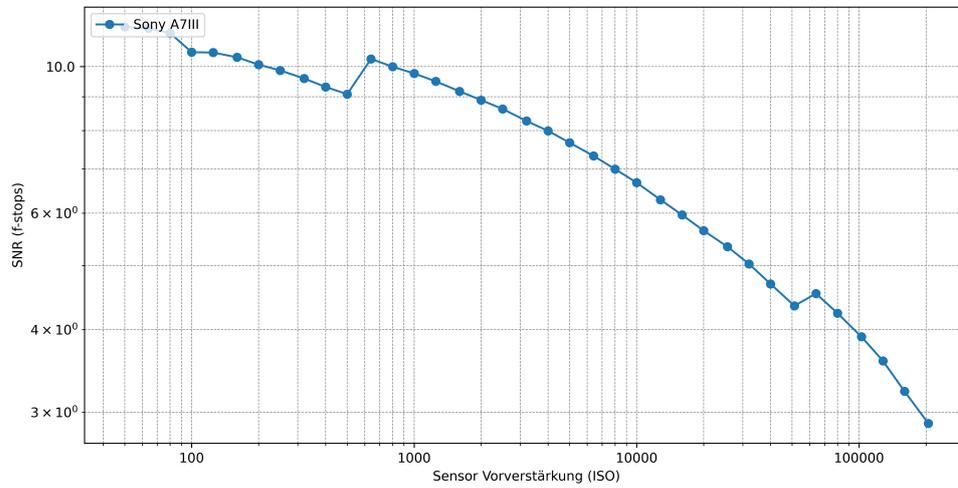


Abbildung C.17.: Signal-Rauschabstand der Sony A7III

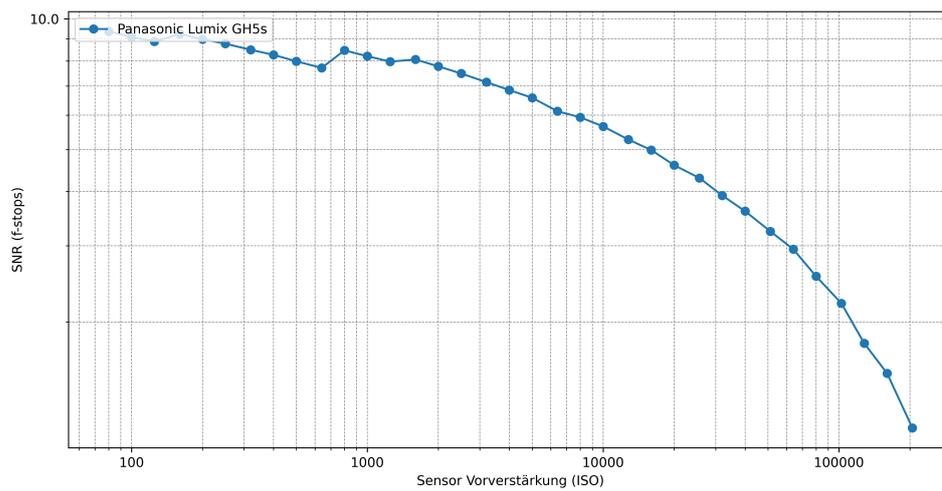


Abbildung C.18.: Signal-Rauschabstand der Lumix GH5s

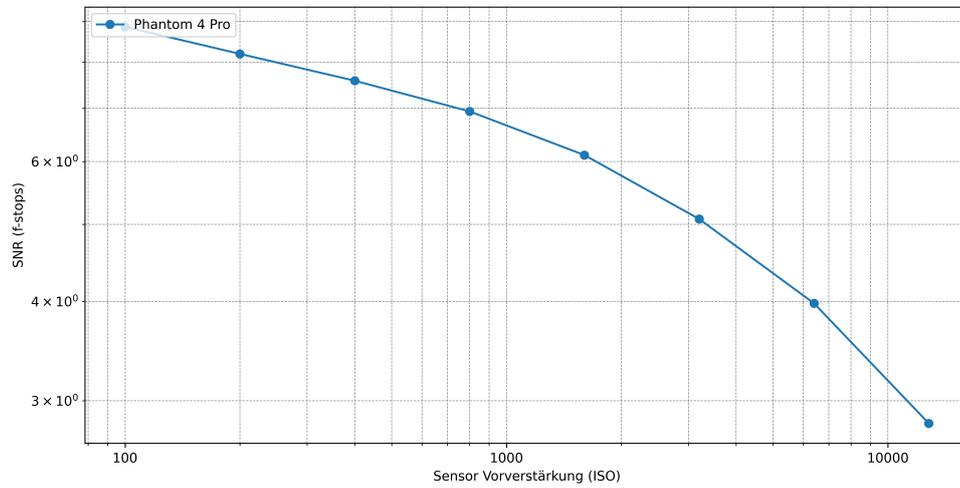


Abbildung C.19.: Signal-Rauschabstand der Phantom 4 Pro

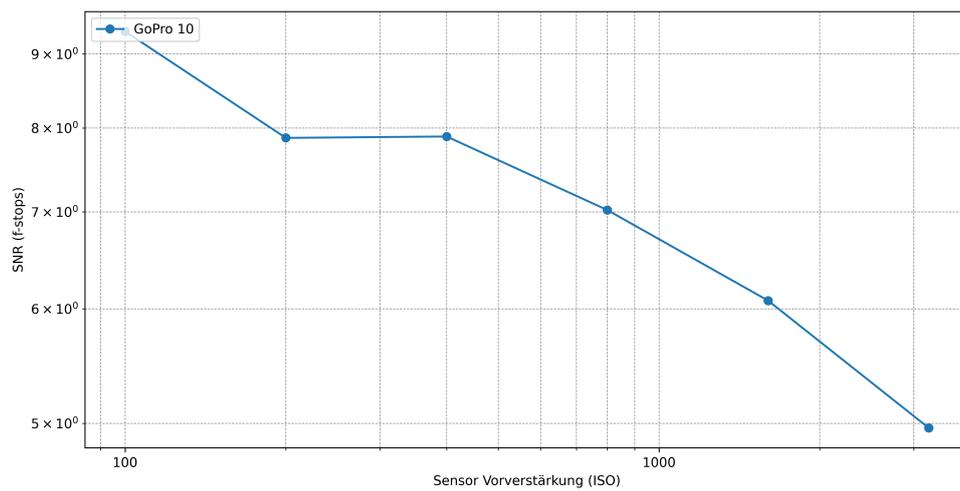


Abbildung C.20.: Signal-Rauschabstand der GoPro 10

C.5. Auswertung 9

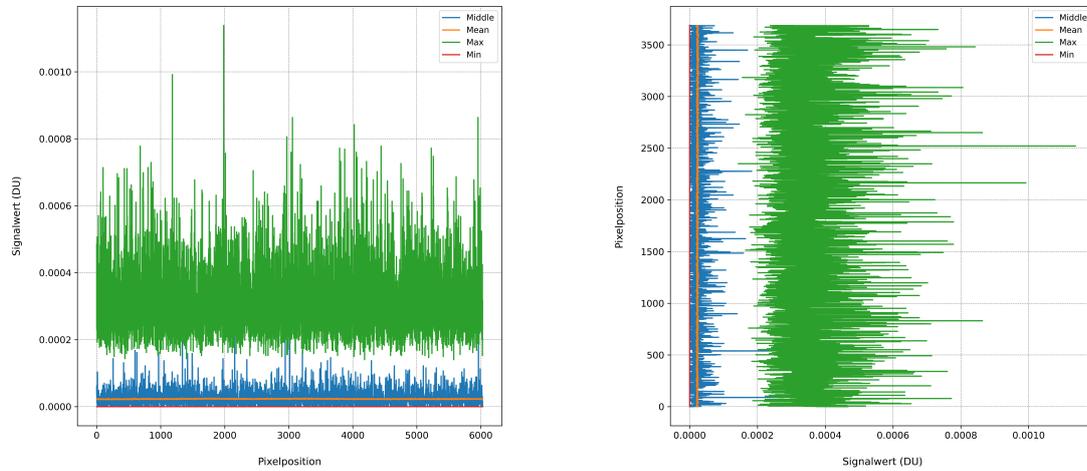


Abbildung C.21.: DSNU der Sony A7IV

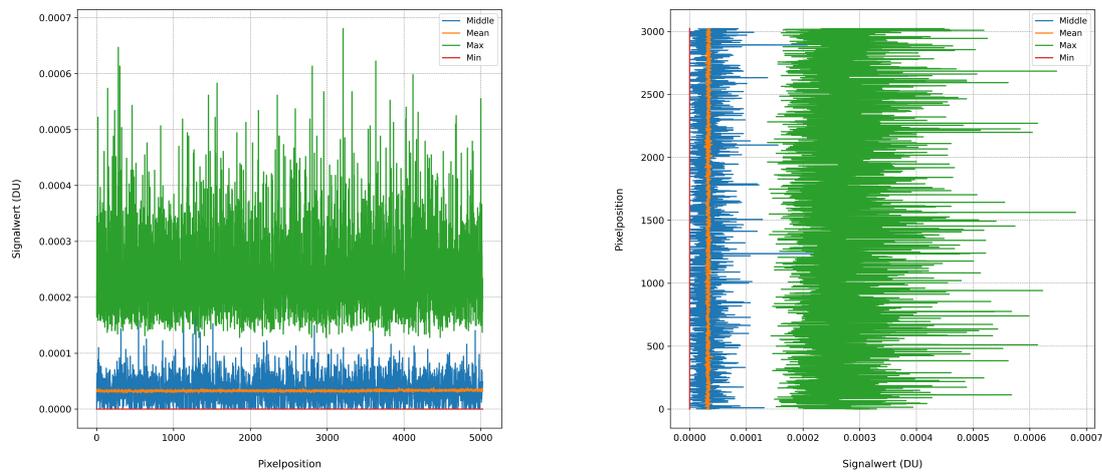


Abbildung C.22.: DSNU der Sony A7III

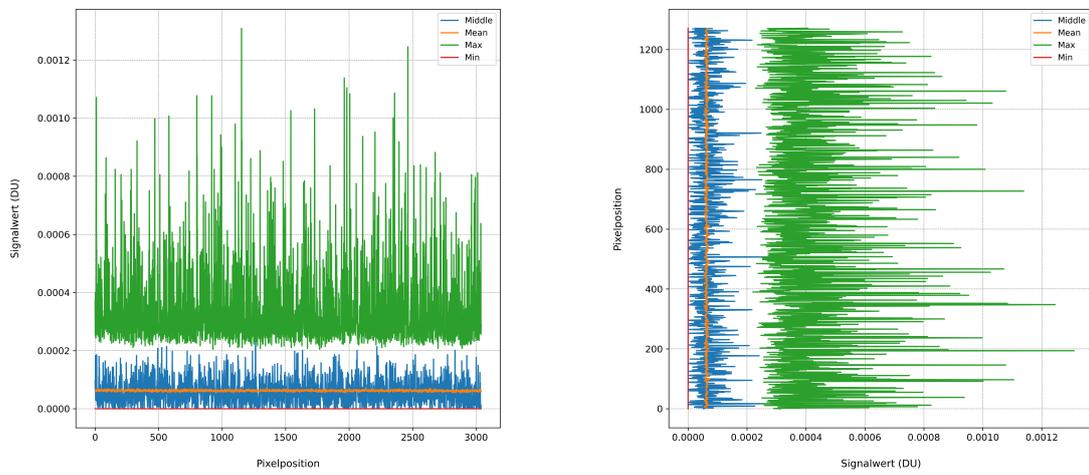


Abbildung C.23.: DSNU der Panasonic Lumix GH5s

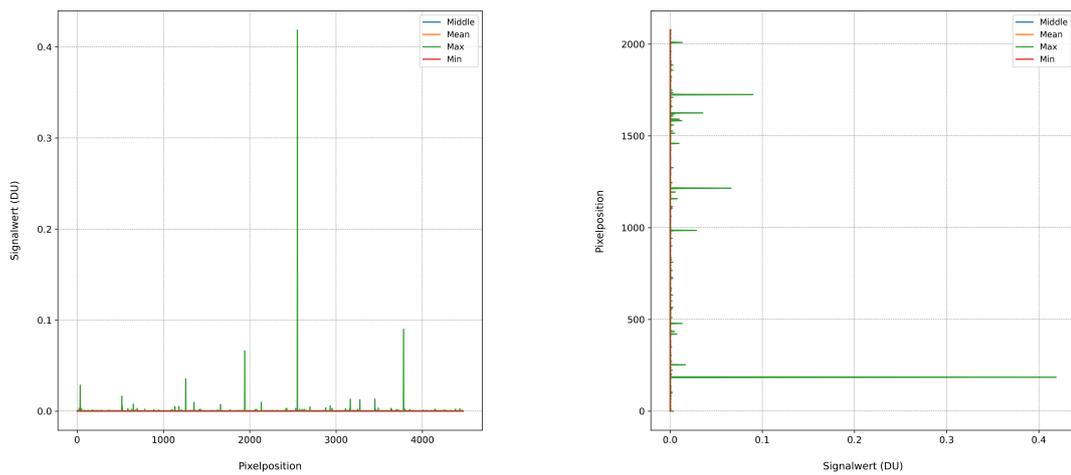


Abbildung C.24.: DSNU der Phantom 4 Pro

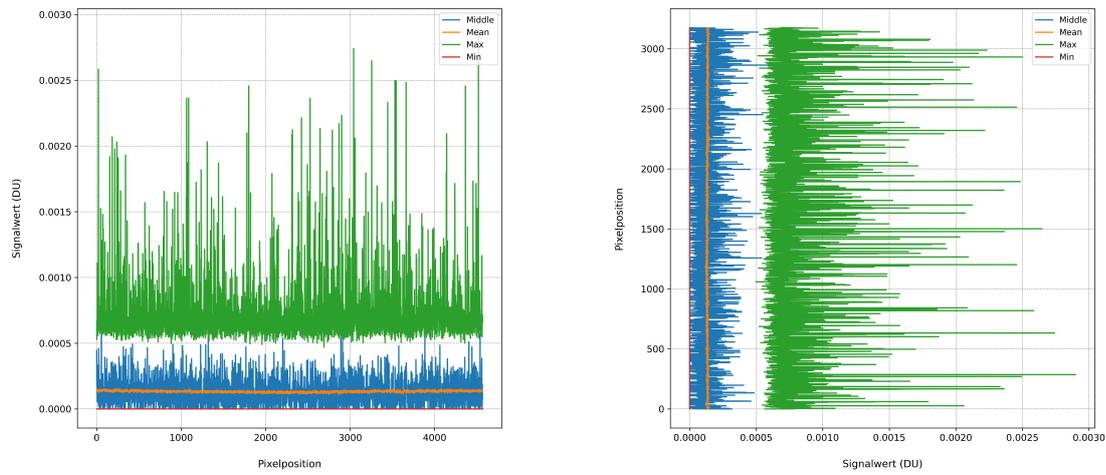


Abbildung C.25.: DSNU der GoPro 10

C.6. Auswertung 10

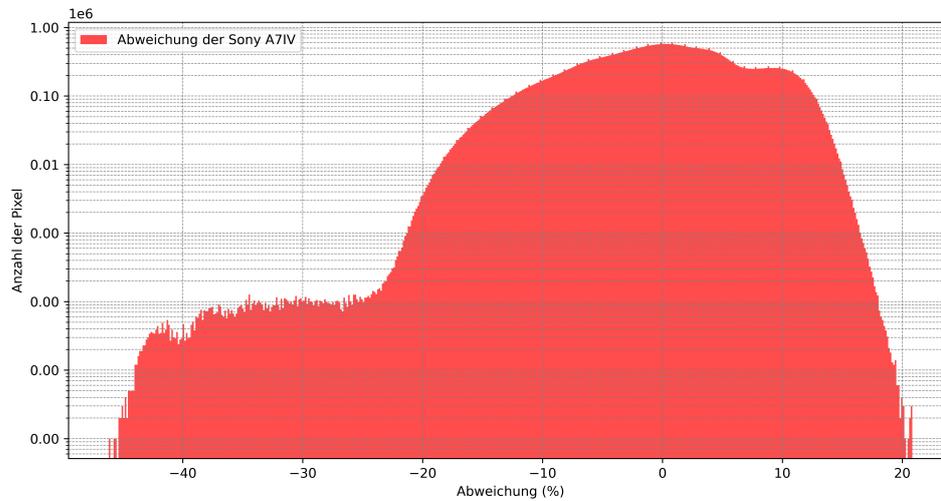


Abbildung C.26.: PRNU der Sony A7IV

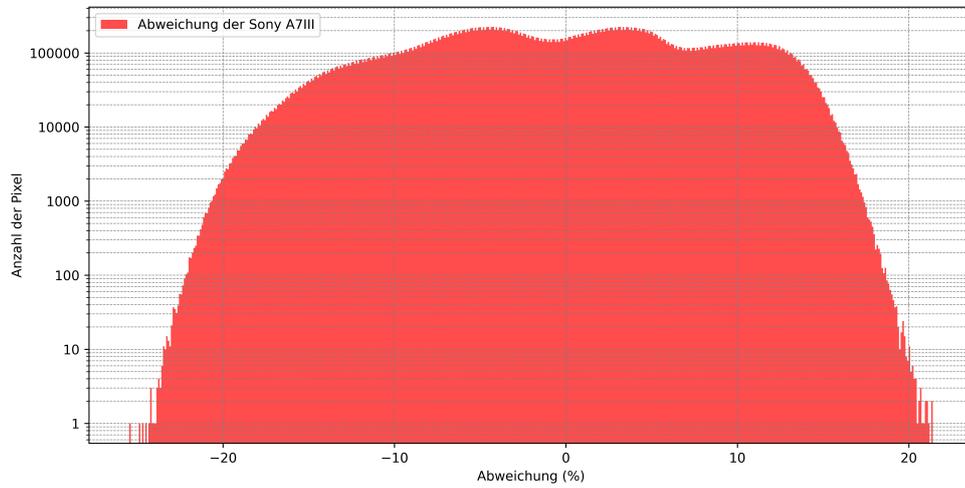


Abbildung C.27.: PRNU der Sony A7III

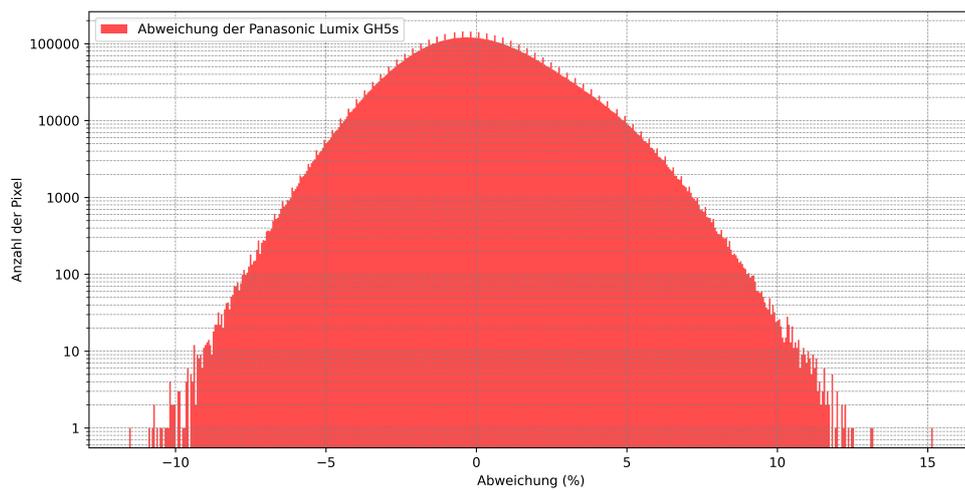


Abbildung C.28.: PRNU der Lumix GH5s

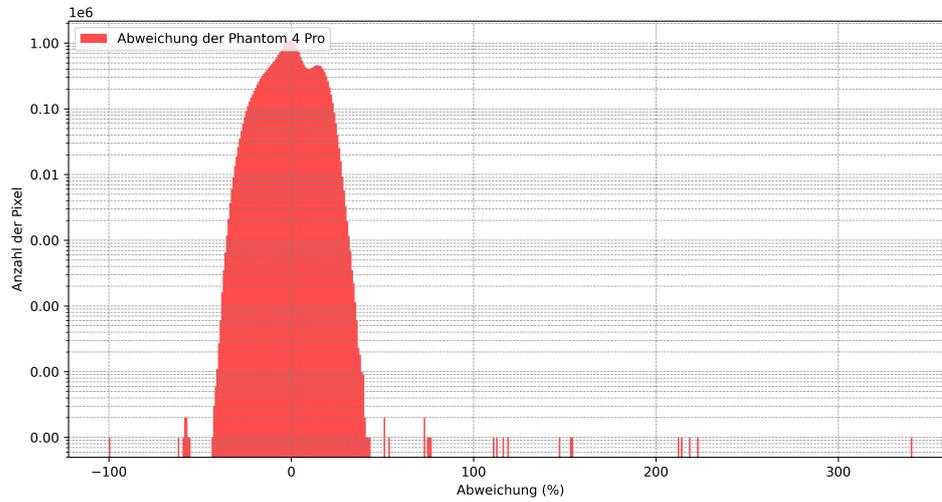


Abbildung C.29.: PRNU der Phantom 4 Pro

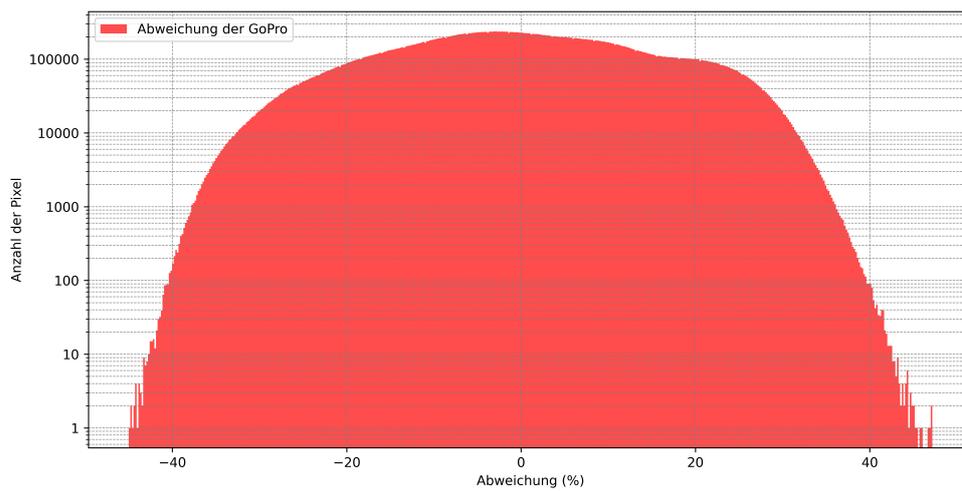


Abbildung C.30.: PRNU der GoPro 10

D. Anhang D: Quellcode

D.1. Quellcode Auswertung 1

```
1
2     import colour
3     import numpy as np
4     import pandas as pd
5
6
7     import matplotlib
8     matplotlib.use('TkAgg')
9     from matplotlib import pyplot as plt
10    from matplotlib.ticker import ScalarFormatter
11
12    from PIL import Image
13
14    import colorspacious as cs
15
16    import os
17    import cv2
18
19    import rawpy
20    import imageio
21
22    import random
23
24    #plt.style.use('dark_background')
25
26
27    # Messdaten der Helligkeit des Spektrometers und der Helligkeit
28    # des angegebenen Luxmeters
29    measured_luminance_luxmeter_lin = [5170, 5000, 4800, 4600,
30    4400, 4200, 4000, 3800, 3600, 3400, 3200, 3000, 2800, 2600,
31    2400, 2200, 2000, 1800, 1600, 1400, 1200, 1000, 800, 600, 400,
32    200]
33    measured_luminance_spektrometer_lin_1 = [2809, 2709, 2603,
34    2490, 2388, 2276, 2177, 2081, 1978, 1872, 1764, 1654, 1550,
35    1443, 1339, 1234, 1127, 1021, 908.2, 794.6, 682.6, 566.2, 450.8,
36    334.9, 222.4, 101.5]
37    measured_luminance_spektrometer_lin_2 = [2811, 2710, 2603,
38    2489, 2388, 2277, 2178, 2081, 1978, 1873, 1763, 1653, 1551,
39    1442, 1338, 1233, 1128, 1021, 908.3, 795.7, 682.5, 567.4, 450.5,
40    335.2, 222.4, 101.7]
41    measured_luminance_spektrometer_lin_3 = [2796, 2711, 2600,
42    2491, 2382, 2272, 2176, 2079, 1975, 1870, 1762, 1655, 1548,
43    1442, 1336, 1231, 1127, 1019, 912.1, 798.3, 679.5, 569.2, 452.8,
44    336.7, 222.7, 103.3]
```

```
32     measured_luminance_spektrometer_lin_4 = [2794, 2708, 2598,
33     2490, 2385, 2277, 2176, 2079, 1975, 1871, 1764, 1654, 1548,
34     1444, 1336, 1232, 1127, 1019, 909.3, 795.6, 681.7, 565.4, 451.7,
35     336.1, 223.6, 102]
36
37     measured_luminance_spektrometer_lin_mean = (
38     measured_luminance_spektrometer_lin_1 +
39     measured_luminance_spektrometer_lin_2 +
40     measured_luminance_spektrometer_lin_3 +
41     measured_luminance_spektrometer_lin_4)
42
43     # Erstellung einer idealen linearen Kurve
44     ideal_lightsource = [x / max(measured_luminance_luxmeter_lin) *
45     max(measured_luminance_spektrometer_lin_mean) for x in
46     measured_luminance_luxmeter_lin]
47
48     # Abweichung vom Durchschnitt:
49     deviation = [0.00302383, 0.00333582, 0.0033818, 0.00443135,
50     0.0027548, 0.00354602, 0.00087135, 0.00672171, 0.00977361,
51     0.01237313, 0.01342966, 0.01401399, 0.01763766, 0.02058089,
52     0.02478099, 0.03037175, 0.03662086, 0.04221511, 0.04544329,
53     0.04578404, 0.04462906, 0.04292014, 0.03788532, 0.02911079,
54     0.02432119, 0.0608569 ]
55
56     plt.figure(figsize = (12, 6))
57
58     # Plot einer idealen Lichtquelle
59     plt.plot(measured_luminance_luxmeter_lin, ideal_lightsource,
60     color = 'red', linestyle = '--', label = 'Ideale Lichtquelle')
61
62     # Plot der gemessenen Lichtquelle
63     plt.plot(measured_luminance_luxmeter_lin,
64     measured_luminance_spektrometer_lin_1, marker = 'o', color = '
65     cyan', linestyle = '-', label = 'Messung 1')
66     plt.plot(measured_luminance_luxmeter_lin,
67     measured_luminance_spektrometer_lin_2, marker = 'o', color = '
68     yellow', linestyle = '-', label = 'Messung 2')
69     plt.plot(measured_luminance_luxmeter_lin,
70     measured_luminance_spektrometer_lin_3, marker = 'o', color = '
71     orange', linestyle = '-', label = 'Messung 3')
72     plt.plot(measured_luminance_luxmeter_lin,
73     measured_luminance_spektrometer_lin_4, marker = 'o', color = '
74     green', linestyle = '-', label = 'Messung 4')
75
76     plt.grid(which = 'both', linestyle = '--', linewidth = '0.5',
77     color = 'grey')
78
79     plt.xscale('linear')
80     plt.yscale('linear')
81
82     plt.xlabel('Lichtstaerke (Lux)', labelpad = 20, fontsize = 12)
83     plt.ylabel('Leuchtdichte (cd/m^2)', labelpad = 20, fontsize =
84     12)
85
86     plt.tick_params(axis = 'both', which = 'major', labelsize = 11)
87     plt.tick_params(axis = 'both', which = 'minor', labelsize = 10)
```

```

63
64
65 plt.legend(loc='upper left')
66
67 plt.savefig('Export/Messung/M01_M02_Verifikation_Ulbrichtkugel/
M01_Verifikation_Luminanz_linear.png', transparent = True, dpi =
68 600)
69 plt.show()
70
71 average_luminance_linear = np.mean([
measured_luminance_spektrometer_lin_1,
measured_luminance_spektrometer_lin_2,
measured_luminance_spektrometer_lin_3,
measured_luminance_spektrometer_lin_4], axis=0)
72
73 absolut_diff_linear = average_luminance_linear -
ideal_lightsource
74 relative_diff_linear = abs((average_luminance_linear /
ideal_lightsource) - 1) * 100
75
76 absolut_max_linear = max(absolut_diff_linear)
77 relative_max_linear = max(relative_diff_linear)
78

```

Listing D.1: Messung 1: Verifikation der Ulbrichtkugel (Luminanz)

```

1 measured_luminance_luxmeter_log = [5000, 2500, 1250, 625,
312.5, 156.25, 78.125, 39.0625, 19.53125]
2 measured_luminance_spektrometer_log_1 = [2706, 1388, 711.5,
362, 181.3, 88.55, 42.08, 19.96, 9.145]
3 measured_luminance_spektrometer_log_2 = [2705, 1386, 719.8,
361.5, 181.1, 88.73, 41.51, 19.93, 9.047]
4 measured_luminance_spektrometer_log_3 = [2706, 1386, 713.4,
361.8, 181.4, 88.24, 41.7, 20.12, 9.273]
5 measured_luminance_spektrometer_log_4 = [2711, 1391, 712.2,
362.1, 181.3, 88.51, 41.86, 20.18, 9.098]
6
7 # Erstellung einer idealen linearen Kurve
8 max_lux = max(measured_luminance_luxmeter_log)
9 max_cd_m2 = max(measured_luminance_spektrometer_log_1)
10
11 ideal_lightsource = [x / max_lux * max_cd_m2 for x in
measured_luminance_luxmeter_log]
12
13 plt.figure(figsize = (12, 6))
14
15 # Plot einer idealen Lichtquelle
16 plt.plot(measured_luminance_luxmeter_log, ideal_lightsource,
color = 'red', linestyle = '--', label = 'Ideale Lichtquelle')
17
18 # Plot der gemessenen Lichtquelle
19 plt.plot(measured_luminance_luxmeter_log,
measured_luminance_spektrometer_log_1, marker='o', color = 'cyan
', linestyle = '-', label = 'Messung 1')
20 plt.plot(measured_luminance_luxmeter_log,

```

```
measured_luminance_spektrometer_log_2, marker='o', color = '
yellow', linestyle = '-', label = 'Messung 2')
21 plt.plot(measured_luminance_luxmeter_log,
measured_luminance_spektrometer_log_3, marker='o', color = '
orange', linestyle = '-', label = 'Messung 3')
22 plt.plot(measured_luminance_luxmeter_log,
measured_luminance_spektrometer_log_4, marker='o', color = '
green', linestyle = '-', label = 'Messung 4')
23
24 plt.grid(which = 'both', linestyle = '--', linewidth = '0.5',
color = 'gray')
25
26 plt.xscale('log')
27 plt.yscale('log')
28
29 plt.xlabel('Lichtstaerke (Lux)', labelpad = 20, fontsize = 12)
30 plt.ylabel('Leuchtdichte (cd/m^2)', labelpad = 20, fontsize =
12)
31
32 plt.tick_params(axis = 'both', which = 'major', labelsize = 11)
33 plt.tick_params(axis = 'both', which = 'minor', labelsize = 10)
34
35 plt.legend()
36
37 # Achsen als Ganzzahl formatieren
38 ax = plt.gca()
39 ax.xaxis.set_major_formatter(ScalarFormatter())
40 ax.yaxis.set_major_formatter(ScalarFormatter())
41
42 plt.savefig('Export/Messung/M01_M02_Verifikation_Ulbrichtkugel/
M01_Verifikation_Luminanz_log.png', transparent = False, dpi =
600)
43 plt.show()
44
45
46
```

Listing D.2: Messung 2: Verifikation der Ulbrichtkugel (Luminanz)

D.2. Quellcode Auswertung 2

```
1     import colour
2     import numpy as np
3     import pandas as pd
4
5
6     import matplotlib
7     matplotlib.use('TkAgg')
8     from matplotlib import pyplot as plt
9     from matplotlib.ticker import ScalarFormatter
10
11    from PIL import Image
12
13    import colorspacious as cs
14
15    import os
16    import cv2
17
18    import rawpy
19    import imageio
20
21    import random
22
23    #plt.style.use('dark_background')
24
25    measured_luminance_luxmeter_lin = [5170, 5000, 4800, 4600,
4400, 4200, 4000, 3800, 3600, 3400, 3200, 3000, 2800, 2600,
2400, 2200, 2000, 1800, 1600, 1400, 1200, 1000, 800, 600, 400,
200]
26    measured_color_temperatur_lin_1 = [3133, 3134, 3134, 3135,
3136, 3137, 3138, 3139, 3140, 3140, 3141, 3141, 3143,
3143, 3145, 3146, 3146, 3150, 3152, 3147, 3147, 3148, 3149,
3152, 3155]
27    measured_color_temperatur_lin_2 = [3133, 3134, 3135, 3135,
3135, 3138, 3139, 3139, 3139, 3140, 3141, 3141, 3141, 3143,
3144, 3144, 3145, 3147, 3149, 3152, 3147, 3148, 3148, 3149,
3151, 3154]
28    measured_color_temperatur_lin_3 = [3133, 3134, 3134, 3135,
3135, 3137, 3138, 3139, 3139, 3140, 3140, 3141, 3141, 3143,
3144, 3145, 3146, 3147, 3150, 3152, 3148, 3148, 3149, 3151,
3153, 3155]
29    measured_color_temperatur_lin_4 = [3133, 3134, 3135, 3135,
3136, 3137, 3138, 3139, 3139, 3140, 3141, 3140, 3141, 3143,
3144, 3145, 3146, 3147, 3150, 3153, 3147, 3147, 3148, 3150,
3151, 3155]
30
31
32    plt.figure(figsize = (12, 6))
33
34    # Plot der gemessenen Lichtquelle
35    plt.plot(measured_luminance_luxmeter_lin,
measured_color_temperatur_lin_1, marker = 'o', color = 'cyan',
linestyle = '-', label = 'Messung 1')
36    plt.plot(measured_luminance_luxmeter_lin,
```

```
measured_color_temperatur_lin_2, marker = 'o', color = 'yellow',
    linestyle = '-', label = 'Messung 2')
37 plt.plot(measured_luminance_luxmeter_lin,
measured_color_temperatur_lin_3, marker = 'o', color = 'orange',
    linestyle = '-', label = 'Messung 3')
38 plt.plot(measured_luminance_luxmeter_lin,
measured_color_temperatur_lin_4, marker = 'o', color = 'green',
    linestyle = '-', label = 'Messung 4')
39
40 plt.grid(which = 'both', linestyle = '--', linewidth = '0.5',
color = 'white')
41
42 plt.xscale('linear')
43 plt.yscale('linear')
44
45 plt.xlabel('Lichtstaerke (Lux)', labelpad = 20, fontsize = 12)
46 plt.ylabel('Farbtemperatur (K)', labelpad = 20, fontsize = 12)
47
48 plt.tick_params(axis = 'both', which = 'major', labelsize = 11)
49 plt.tick_params(axis = 'both', which = 'minor', labelsize = 10)
50
51 plt.legend()
52
53 plt.savefig('Export/Messung/M01_M02_Verifikation_Ulbrichtkugel/
M02_Verifikation_Farbtemperatur_linear.png', transparent = False
, dpi = 600)
54 plt.show()
55
56
57 transposed_data = [
58     [3133, 3133, 3133, 3133],
59     [3134, 3134, 3134, 3134],
60     [3134, 3135, 3134, 3135],
61     [3135, 3135, 3135, 3135],
62     [3136, 3135, 3135, 3136],
63     [3137, 3138, 3137, 3137],
64     [3138, 3139, 3138, 3138],
65     [3139, 3139, 3139, 3139],
66     [3140, 3139, 3139, 3139],
67     [3140, 3140, 3140, 3140],
68     [3141, 3141, 3140, 3141],
69     [3141, 3141, 3141, 3140],
70     [3141, 3141, 3141, 3141],
71     [3143, 3143, 3143, 3143],
72     [3143, 3144, 3144, 3144],
73     [3145, 3144, 3145, 3145],
74     [3146, 3145, 3146, 3146],
75     [3146, 3147, 3147, 3147],
76     [3150, 3149, 3150, 3150],
77     [3152, 3152, 3152, 3153],
78     [3147, 3147, 3148, 3147],
79     [3147, 3148, 3148, 3147],
80     [3148, 3148, 3149, 3148],
81     [3149, 3149, 3151, 3150],
82     [3152, 3151, 3153, 3151],
83     [3155, 3154, 3155, 3155],
```

```

84     ]
85
86     measured_luminance_luxmeter_lin = [5170, 5000, 4800, 4600,
4400, 4200, 4000, 3800, 3600, 3400, 3200, 3000, 2800, 2600,
2400, 2200, 2000, 1800, 1600, 1400, 1200, 1000, 800, 600, 400,
200]
87
88     plt.figure(figsize=(14, 8))
89     plt.boxplot(transposed_data, positions =
measured_luminance_luxmeter_lin, widths = 150)
90
91     # plt.title('Boxplot der Farbtemperaturmessungen nach Index')
92     plt.xlabel('Lichtstaerke (Lux)', labelpad = 20, fontsize = 12)
93     plt.ylabel('Farbtemperatur (K)', labelpad = 20, fontsize = 12)
94
95     xticks_positions = measured_luminance_luxmeter_lin[:5]
96     xticks_labels = [str(val) for val in xticks_positions]
97
98     plt.xticks(ticks=xticks_positions, labels=xticks_labels)
99
100    plt.tick_params(axis = 'both', which = 'major', labelsize = 11)
101    plt.tick_params(axis = 'both', which = 'minor', labelsize = 10)
102
103    # Anzeigen des Plots
104    plt.grid(which = 'both', linestyle = '--', linewidth = 0.5)
105
106    plt.savefig('Export/Messung/M01_M02_Verifikation_Ulbrichtkugel/
M02_Verifikation_Farbtemperatur_linear_boxplot.png', transparent
= False, dpi = 600)
107    plt.show()
108
109

```

Listing D.3: Messung 1: Verifikation der Ulbrichtkugel (Farbtemperatur)

```

1     measured_luminance_luxmeter_log =
measured_luminance_luxmeter_log = [5000, 2500, 1250, 625, 312.5,
156.25, 78.125, 39.0625, 19.53125]
2     measured_color_temperatur_log_1 = [3134, 3143, 3153, 3149,
3154, 3159, 3157, 3132, 3105]
3     measured_color_temperatur_log_2 = [3134, 3143, 3153, 3149,
3151, 3157, 3154, 3129, 3105]
4     measured_color_temperatur_log_3 = [3134, 3144, 3153, 3151,
3154, 3158, 3156, 3130, 3102]
5     measured_color_temperatur_log_4 = [3134, 3144, 3155, 3150,
3153, 3161, 3157, 3132, 3103]
6
7     plt.figure(figsize=(12, 6))
8
9     # Plot der gemessenen Lichtquelle
10    plt.plot(measured_luminance_luxmeter_log,
measured_color_temperatur_log_1, marker = 'o', color = 'cyan',
linestyle = '-', label = 'Messung 1')
11    plt.plot(measured_luminance_luxmeter_log,
measured_color_temperatur_log_2, marker = 'o', color = 'yellow',
linestyle = '-', label = 'Messung 2')

```

```
12     plt.plot(measured_luminance_luxmeter_log ,
13             measured_color_temperatur_log_3, marker = 'o', color = 'orange',
14             linestyle = '-', label = 'Messung 3')
15     plt.plot(measured_luminance_luxmeter_log ,
16             measured_color_temperatur_log_4, marker = 'o', color = 'green',
17             linestyle = '-', label = 'Messung 4')
18
19     plt.grid(which = 'both', linestyle = '--', linewidth = '0.5',
20             color = 'gray')
21
22     plt.xscale('log')
23     plt.yscale('linear')
24
25     plt.xlabel('Lichtstaerke (Lux)', labelpad = 20, fontsize = 12)
26     plt.ylabel('Farbtemperatur (K)', labelpad = 20, fontsize = 12)
27
28     plt.tick_params(axis = 'both', which = 'major', labelsize = 11)
29     plt.tick_params(axis = 'both', which = 'minor', labelsize = 10)
30
31     plt.legend()
32
33     # Achsen als Ganzzahl formatieren
34     ax = plt.gca()
35     ax.xaxis.set_major_formatter(ScalarFormatter())
36     ax.yaxis.set_major_formatter(ScalarFormatter())
37
38     plt.savefig('Export/Messung/M01_M02_Verifikation_Ulbrichtkugel/
39     M02_Verifikation_Farbtemperatur_log.png', transparent = False,
40     dpi = 600)
41     plt.show()
42
```

Listing D.4: Messung 2: Verifikation der Ulbrichtkugel (Farbtemperatur)

D.3. Quellcode Auswertung 3

```
1     import colour
2     import numpy as np
3     import pandas as pd
4
5
6     import matplotlib
7     matplotlib.use('TkAgg')
8     from matplotlib import pyplot as plt
9     from matplotlib.ticker import ScalarFormatter
10
11    from PIL import Image
12
13    import colorspacious as cs
14
15    import os
16    import cv2
17
18    import rawpy
19    import imageio
20
21    import random
22
23    #plt.style.use('dark_background')
24
25    # Messdaten und Herstellerangaben
26    measured_luminance_spektrometer_chart = [2045, 1801, 1510,
27    1253, 982.8, 769, 571.9, 405.1, 271.6, 173.2, 95.29, 38.71,
28    10.61]
29    density_hersteller = [0.05, 0.13, 0.21, 0.30, 0.40, 0.51, 0.63,
30    0.78, 0.95, 1.15, 1.41, 1.77, 2.35]
31
32    # Maximale Luminanz bestimmen
33    L0 = max(measured_luminance_spektrometer_chart)/0.89
34
35    # Berechnung der Dichtewerte
36    calculated_density = [np.log10(L0 / L) for L in
37    measured_luminance_spektrometer_chart]
38
39
40    plt.figure(figsize = (12, 6))
41
42    # Vergleich der berechneten optischen Dichtewerte mit den
43    Herstellerangaben
44    plt.plot(calculated_density, label = 'gemessene optische Dichte',
45    color = 'blue')
46    plt.plot(density_hersteller, label = 'optische Dichte laut
47    Herstellerangaben', linestyle = '--', color = 'red')
48
49    plt.grid(which = 'both', linestyle = '--', linewidth = '0.5',
50    color = 'gray')
51
52    plt.xscale('linear')
53    plt.yscale('linear')
```

```
46
47     #plt.title('Vergleich der errechneten optischen Dichte mit den
48     Herstellerangaben zum TE223')
49     plt.xlabel('Messpunkte', labelpad = 20, fontsize = 12)
50     plt.ylabel('optische Dichte', labelpad = 20, fontsize = 12)
51
52     plt.tick_params(axis = 'both', which = 'major', labelsize = 11)
53     plt.tick_params(axis = 'both', which = 'minor', labelsize = 10)
54
55     plt.legend()
56
57     plt.savefig('Export/Messung/M03_Verifikation_TE233/
58     M03_optische_Dichte_TE223.png', transparent = False, dpi = 600)
59     plt.show()
```

Listing D.5: Verifikation TE223

D.4. Quellcode Auswertung 4

```
1 import colour
2 import numpy as np
3 import pandas as pd
4
5
6 import matplotlib
7 matplotlib.use('TkAgg')
8 from matplotlib import pyplot as plt
9 from matplotlib.ticker import ScalarFormatter
10
11 from PIL import Image
12
13 import colorspacious as cs
14
15 import os
16 import cv2
17
18 import rawpy
19 import imageio
20
21 import random
22
23 #plt.style.use('dark_background')
24
25
26 # Gemessenen Colorpatches
27 measured_color_patches = pd.read_csv('Assets/Messung/
M04_Verifikation_TE188/M04_Verifikation_TE188.csv', sep = ';')
28
29 # XYZ Werte transformieren in D65
30 illuminant_measurement = np.array([0.4298, 0.4074])
31 illuminant_A = [0.4476, 0.4074]
32 illuminant_D65 = np.array([0.31382, 0.33100])
33
34 value_X = []
35 value_Y = []
36 value_Z = []
37
38 value_L = []
39 value_a = []
40 value_b = []
41
42 value_x = []
43 value_y = []
44
45 for index, row in measured_color_patches.iterrows():
46     R, G, B = colour.models.XYZ_to_RGB([row['X'], row['Y'], row
['Z']], "ITU-R BT.2020", illuminant_measurement, "Bradford")
47     X, Y, Z = np.round(colour.models.RGB_to_XYZ([R, G, B], "ITU
-R BT.2020", illuminant_D65, "Bradford") / 100, 3)
48     L, a, b = colour.models.XYZ_to_Lab([X, Y, Z])
49     x, y = np.round(colour.models.XYZ_to_xy([X, Y, Z]), 2)
50
```

```

51     value_X.append(X)
52     value_Y.append(Y)
53     value_Z.append(Z)
54
55     value_L.append(L)
56     value_a.append(a)
57     value_b.append(b)
58
59     value_x.append(x)
60     value_y.append(y)
61
62     measured_color_patches['X'] = value_X
63     measured_color_patches['Y'] = value_Y
64     measured_color_patches['Z'] = value_Z
65
66     measured_color_patches['L'] = value_L
67     measured_color_patches['a'] = value_a
68     measured_color_patches['b'] = value_b
69
70     measured_color_patches['x'] = value_x
71     measured_color_patches['y'] = value_y
72
73     reference_color_patches = pd.DataFrame(
74         {
75             "Color": ['Dark Skin', 'Light Skin', 'Blue Sky', '
Foliage', 'Blue Flower', 'Bluish Green', 'Orange', 'Purplish
blue', 'moderate red', 'purple', 'yellow green', 'orange yellow'
, 'blue flower', 'green', 'red', 'yellow', 'magenta', 'cyan', '
White', 'Neutral', 'Neutral', 'Neutral', 'Neutral', 'Black'],
76             "X": [14.58, 54.4, 15.34, 7.14, 49.31, 41.71, 47.89,
18.23, 40.32, 13.53, 60.19, 53.34, 8.6, 9.97, 37.11, 55.78,
43.54, 47.09, 95.05, 68.99, 41.54, 22.53, 11.07, 3.12],
77             "Y": [12.51, 49.35, 17.61, 12.41, 45.51, 57.74, 37.39,
18.27, 28.32, 8.25, 82.73, 45.59, 3.75, 23.31, 19.08, 63.14,
21.44, 59.26, 100, 72.74, 43.83, 23.77, 11.68, 3.3],
78             "Z": [8.65, 37.85, 30.46, 3.2, 85.69, 67.46, 6.82,
64.25, 18.75, 38.88, 33.24, 8.43, 45.32, 6.4, 2.55, 5.06, 52.53,
102.19, 108.88, 78.32, 46.9, 25.4, 12.37, 3.48],
79             "x": [0.41, 0.38, 0.24, 0.31, 0.27, 0.25, 0.52, 0.18,
0.46, 0.22, 0.34, 0.5, 0.15, 0.25, 0.63, 0.45, 0.37, 0.23, 0.31,
0.31, 0.31, 0.31, 0.32, 0.32],
80             "y": [0.35, 0.35, 0.28, 0.55, 0.25, 0.35, 0.41, 0.18,
0.32, 0.14, 0.47, 0.42, 0.07, 0.59, 0.32, 0.51, 0.18, 0.28,
0.33, 0.33, 0.33, 0.33, 0.33, 0.33]
81         }
82     )
83
84     #Normieren der XYZ-Werte
85
86     value_X_norm = []
87     value_Y_norm = []
88     value_Z_norm = []
89
90     for index, row in measured_color_patches.iterrows():
91         value_X_norm.append(row['X'] * (1 / row['Y']))
92         value_Y_norm.append(1)

```

```
93     value_Z_norm.append(row['Z'] * (1 / row['Y']))
94
95     measured_color_patches['X'] = value_X_norm
96     measured_color_patches['Y'] = value_Y_norm
97     measured_color_patches['Z'] = value_Z_norm
98
99
100     value_X_norm = []
101     value_Y_norm = []
102     value_Z_norm = []
103
104     for index, row in reference_color_patches.iterrows():
105         value_X_norm.append(row['X'] * (1 / row['Y']))
106         value_Y_norm.append(1)
107         value_Z_norm.append(row['Z'] * (1 / row['Y']))
108
109     reference_color_patches['X'] = value_X_norm
110     reference_color_patches['Y'] = value_Y_norm
111     reference_color_patches['Z'] = value_Z_norm
112
113
114
115     # La*b* Werte berechnen
116     value_L = []
117     value_a = []
118     value_b = []
119
120     for index, row in measured_color_patches.iterrows():
121         L, a, b = colour.models.XYZ_to_Lab([row['X'], row['Y'], row
122 ['Z']])
123         value_L.append(L)
124         value_a.append(a)
125         value_b.append(b)
126
127     measured_color_patches['L'] = value_L
128     measured_color_patches['a'] = value_a
129     measured_color_patches['b'] = value_b
130
131
132     value_L = []
133     value_a = []
134     value_b = []
135
136     for index, row in reference_color_patches.iterrows():
137         L, a, b = colour.models.XYZ_to_Lab([row['X'], row['Y'], row
138 ['Z']])
139         value_L.append(L)
140         value_a.append(a)
141         value_b.append(b)
142
143     reference_color_patches['L'] = value_L
144     reference_color_patches['a'] = value_a
145     reference_color_patches['b'] = value_b
146
147     delta_e = []
148     for i in range(0,24):
```

```
147
148     a = [reference_color_patches['L'][i],
reference_color_patches['a'][i], reference_color_patches['b'][i
]]
149     b = [measured_color_patches['L'][i], measured_color_patches
['a'][i], measured_color_patches['b'][i]]
150
151     delta_e.append(colour.delta_E(a ,b))
152
153
154     fig = plt.figure(figsize = (12, 6))
155
156     ax = fig.add_subplot(111, projection='3d')
157
158     ax.scatter(reference_color_patches['a'],
reference_color_patches['b'], reference_color_patches['L'],
color = 'r', s = 50, label='La*b* Points Reference')
159     ax.scatter(measured_color_patches['a'], measured_color_patches[
'b'], measured_color_patches['L'], color = 'b', s = 50, label='
La*b* Points Measured')
160
161     ax.set_xlabel('a*', labelpad = 20, fontsize = 12)
162     ax.set_ylabel('b*', labelpad = 20, fontsize = 12)
163     ax.set_zlabel('L', labelpad = 20, fontsize = 12)
164
165     ax.tick_params(axis = 'both', which = 'major', labelsize = 11)
166     ax.tick_params(axis = 'both', which = 'minor', labelsize = 10)
167
168
169     ax.legend(loc = 'upper left')
170
171     plt.savefig('Export/Messung/M04_Verifikation_TE188/
M04_Verifikation_TE188.png', transparent = False, dpi = 600)
172     plt.show()
173
174
175
```

Listing D.6: Verifikation TE188

D.5. Quellcode Auswertung 5

```
1     import colour
2     import numpy as np
3     import pandas as pd
4
5
6     import matplotlib
7     matplotlib.use('TkAgg')
8     from matplotlib import pyplot as plt
9     from matplotlib.ticker import ScalarFormatter
10
11    from PIL import Image
12
13    import colorspacious as cs
14
15    import os
16    import cv2
17
18    import rawpy
19    import imageio
20
21    import random
22
23    #plt.style.use('dark_background')
24
25
26    # Funktion zum einlesen von 8 bit Bildern
27    def read_image_8bit(file_path):
28        with rawpy.imread(file_path) as raw:
29            rgb_image = raw.postprocess(
30                use_camera_wb = True,
31                #no_auto_bright = True,
32                output_bps = 8
33            )
34        rgb_image = rgb_image / 256
35        return rgb_image
36
37    # Funktion zum einlesen von 16 bit Bildern
38    def read_image_16bit(file_path):
39
40        file_ending = file_path.split('.')[-1].lower()
41
42        if file_ending in ['nef', 'cr2', 'arw', 'dng', 'raw', 'rw2'
43    ]:
44            with rawpy.imread(file_path) as raw:
45                rgb_image = raw.postprocess(
46                    use_camera_wb = True,
47                    no_auto_bright = True,
48                    output_bps = 16,
49                    # output_color
50                    gamma = (1, 1)
51                )
52            rgb_image = rgb_image / 65535
```

```

53     elif file_ending in ['.tif', '.tiff']:
54         rgb_image = Image.open(file_path)
55         rgb_image = np.array(rgb_image, dtype = np.float32)
56         rgb_image = rgb_image / 255
57
58     return rgb_image
59
60 def normalize_image(image):
61     return image / image.max()
62
63 def crop_image(image, x1, y1, x2, y2):
64     return image[int(y1) : int(y2), int(x1) : int(x2)]
65
66 def list_file_names(link: str):
67     all_files = os.listdir(link)
68     raw_files = [file for file in all_files]
69     raw_files.sort()
70
71     return raw_files
72
73 def crop_sides(img, crop_size):
74     height = img.shape[0]
75     width = img.shape[1]
76     top = crop_size
77     bottom = height - crop_size
78     left = crop_size
79     right = width - crop_size
80     cropped_img = img[top:bottom, left:right]
81     return cropped_img
82
83 def measure_xy_positions(img):
84     preview = Linear_to_sRGB(img)
85
86     plt.imshow(preview, cmap = 'gray')
87     plt.title('Klicken Sie auf die Eckpunkte der Patches, um
88 einen Bereich auszuwaehlen')
89     points = plt.ginput(n = 26, timeout = 1000)
90     plt.close()
91
92     return points
93
94 def measure_gammacurve(image_directory_signal: str):
95     file_list_signal = list_file_names(image_directory_signal)
96
97     signal_images = []
98     mean_images = []
99
100     for path in file_list_signal:
101         img = read_image_16bit(f'{image_directory_signal}/{path
102 }')
103         #img = normalize_image(img)
104
105         signal_images.append((img))
106
107     points = measure_xy_positions(signal_images[2])
108     (x1, y1), (x2, y2), (x3, y3), (x4, y4), (x5, y5), (x6, y6),

```

```
(x7, y7), (x8, y8), (x9, y9), (x10, y10), (x11, y11), (x12, y12
), (x13, y13), (x14, y14), (x15, y15), (x16, y16), (x17, y17), (
x18, y18), (x19, y19), (x20, y20), (x21, y21), (x22, y22), (x23,
y23), (x24, y24), (x25, y25), (x26, y26) = points
107
108     for img in signal_images:
109         cropped_image_1 = crop_image(img, x1, y1, x2, y2)
110         mean_images.append(np.mean(cropped_image_1))
111
112         cropped_image_2 = crop_image(img, x3, y3, x4, y4)
113         mean_images.append(np.mean(cropped_image_2))
114
115         cropped_image_3 = crop_image(img, x5, y5, x6, y6)
116         mean_images.append(np.mean(cropped_image_3))
117
118         cropped_image_4 = crop_image(img, x7, y7, x8, y8)
119         mean_images.append(np.mean(cropped_image_4))
120
121         cropped_image_5 = crop_image(img, x9, y9, x10, y10)
122         mean_images.append(np.mean(cropped_image_5))
123
124         cropped_image_6 = crop_image(img, x11, y11, x12, y12)
125         mean_images.append(np.mean(cropped_image_6))
126
127         cropped_image_7 = crop_image(img, x13, y13, x14, y14)
128         mean_images.append(np.mean(cropped_image_7))
129
130         cropped_image_8 = crop_image(img, x15, y15, x16, y16)
131         mean_images.append(np.mean(cropped_image_8))
132
133         cropped_image_9 = crop_image(img, x17, y17, x18, y18)
134         mean_images.append(np.mean(cropped_image_9))
135
136         cropped_image_10 = crop_image(img, x19, y19, x20, y20)
137         mean_images.append(np.mean(cropped_image_10))
138
139         cropped_image_11 = crop_image(img, x21, y21, x22, y22)
140         mean_images.append(np.mean(cropped_image_11))
141
142         cropped_image_12 = crop_image(img, x23, y23, x24, y24)
143         mean_images.append(np.mean(cropped_image_12))
144
145         cropped_image_13 = crop_image(img, x25, y25, x26, y26)
146         mean_images.append(np.mean(cropped_image_13))
147
148     return signal_images, mean_images
149
150     measured_luminance_spektrometer_5000_1600 = [26464, 23200,
19616, 16224, 12825.6, 10211.2, 7424, 5283.2, 3560.0, 2292.8,
1265.12, 509.92, 120.653333328]
151     measured_luminance_spektrometer_5000_800 = [13232, 11600, 9808,
8112, 6412.8, 5105.6, 3712, 2641.6, 1780.0, 1146.4, 632.56,
254.96, 60.326666664]
152     measured_luminance_spektrometer_5000_400 = [6616, 5800, 4904,
4056, 3206.4, 2552.8, 1856, 1320.8, 890.0, 573.2, 316.28,
127.48, 30.163333332]
```

```

153     measured_luminance_spektrometer_5000_200 = [3308, 2900, 2452,
2028, 1603.2, 1276.4, 928, 660.4, 445.0, 286.6, 158.14, 63.74,
15.081666666]
154     measured_luminance_spektrometer_5000 = [1654, 1450, 1226, 1014,
801.6, 638.2, 464, 330.2, 222.5, 143.3, 79.07, 31.87,
7.540833333]
155     measured_luminance_spektrometer_2500 = [890.8, 748, 632.5,
520.1, 413.5, 325.8, 240.7, 172.4, 114.7, 73.43, 40.82, 17.21,
3.770416667]
156     measured_luminance_spektrometer_1250 = [452.3, 382, 320.3, 264,
211.7, 161.4, 120.7, 86.62, 58.26, 37.21, 20.72, 8.167,
1.885208333]
157     measured_luminance_spektrometer_625 = [222.6, 186.6, 156.4,
129.1, 102, 78.92, 59.21, 42.39, 28.31, 18.36, 10.16, 4.0835,
0.942604167]
158     measured_luminance_spektrometer_312 = [109.6, 92.1, 78.24,
63.91, 50.62, 38.94, 28.88, 20.84, 13.92, 8.98, 5.08, 2.04175,
0.471302083]
159     measured_luminance_spektrometer_156 = [53.92, 45.07, 38.21,
32.21, 24.81, 19.13, 14.33, 10.31, 6.96, 4.49, 2.54, 1.020875,
0.235651042]
160     measured_luminance_spektrometer_78 = [26.5, 22.19, 18.82,
15.38, 12.23, 9.756, 7.165, 5.155, 3.48, 2.245, 1.27, 0.5104375,
0.117825521]
161     measured_luminance_spektrometer_39 = [12.7, 10.79, 9.12, 7.512,
6.115, 4.878, 3.5825, 2.5775, 1.74, 1.1225, 0.635, 0.25521875,
0.05891276]
162     measured_luminance_spektrometer_19 = [6.363, 5.395, 4.56,
3.756, 3.0575, 2.439, 1.79125, 1.28875, 0.87, 0.56125, 0.3175,
0.127609375, 0.02945638]
163
164     measured_luminance_spektrometer_all = (
measured_luminance_spektrometer_5000_800 +
measured_luminance_spektrometer_5000_400 +
measured_luminance_spektrometer_5000_200 +
measured_luminance_spektrometer_5000 +
measured_luminance_spektrometer_2500 +
measured_luminance_spektrometer_1250 +
measured_luminance_spektrometer_625 +
measured_luminance_spektrometer_312 +
measured_luminance_spektrometer_156 +
measured_luminance_spektrometer_78 +
measured_luminance_spektrometer_39 +
measured_luminance_spektrometer_19)
165
166
167     def plot_gammacurve(mean, luminance, full_well_capacity, label:
str, link: str):
168         plt.figure(figsize = (12, 6))
169
170         reference_x = np.linspace(0, 1, 200000)
171         reference_y = (colour.models.oetf_inverse_BT2100_HLG(
reference_x)**0.5) * full_well_capacity
172         reference_y2 = (colour.models.oetf_inverse_BT709(
reference_x)**0.5) * full_well_capacity
173         reference_y3 = reference_x * full_well_capacity

```

```
174
175
176     plt.plot(mean, luminance, marker = 'o', linestyle='none',
label = label)
177     plt.plot(reference_x, reference_y, linestyle='--', label = '
BT.2100 HLG Referenz OETF', color = 'red')
178     plt.plot(reference_x, reference_y2, linestyle='--', label =
'BT.709 Referenz OETF', color = 'green')
179     plt.plot(reference_x, reference_y3, linestyle='--', label =
'Linear Referenz', color = 'orange')
180
181     plt.axhline(y = full_well_capacity, color = 'green',
linestyle = '--', linewidth = 1, label = 'Full Well Capacity')
182     plt.axhline(y = 7, color = 'red', linestyle = '--',
linewidth = 1, label = 'Messgrenze des Spektrometers')
183
184     plt.grid(which = 'both', linestyle = '--', linewidth = '0.5
', color = 'grey')
185
186     plt.xscale('log')
187     plt.yscale('log')
188
189     plt.xlabel('Signalwerte', labelpad = 20, fontsize = 12)
190     plt.ylabel('Luminanz in cd/m^2', labelpad = 20, fontsize =
12)
191
192     plt.tick_params(axis = 'both', which = 'major', labelsize =
11)
193     plt.tick_params(axis = 'both', which = 'minor', labelsize =
10)
194
195     plt.xlim(0, 2)
196
197     plt.legend(loc='upper left')
198
199     plt.savefig(link, transparent = False, dpi = 600)
200     plt.show()
201
202     # Sony A7IV
203     sonya7IV_signal, sonya7IV_mean = measure_gammacurve(
image_directory_signal = 'Assets/Messung/M05_Gammacurve/Sony
A7IV')
204     measured_luminance_spektrometer_sonya7IV = (
measured_luminance_spektrometer_5000_800 +
measured_luminance_spektrometer_5000_400 +
measured_luminance_spektrometer_5000_200 +
measured_luminance_spektrometer_5000 +
measured_luminance_spektrometer_2500 +
measured_luminance_spektrometer_1250 +
measured_luminance_spektrometer_625 +
measured_luminance_spektrometer_312 +
measured_luminance_spektrometer_156 +
measured_luminance_spektrometer_78 +
measured_luminance_spektrometer_39 +
measured_luminance_spektrometer_19)
205     full_well_capacity_a7IV = [a / b for a, b in zip(
```

```
measured_luminance_spektrometer_sonya7IV, sonya7IV_mean)]
206     full_well_capacity_a7IV = np.mean(np.array(
full_well_capacity_a7IV))
207     plot_gammapcurve(sonya7IV_mean,
measured_luminance_spektrometer_sonya7IV,
full_well_capacity_a7IV, 'Sony A7IV', 'Export/Messung/
M05_Gammapcurve/Luminanz_SonyA7IV.png')
208
209     # Sony A7III
210     sonya7III_signal, sonya7III_mean = measure_gammapcurve(
image_directory_signal = 'Assets/Messung/M05_Gammapcurve/Sony
A7III')
211     measured_luminance_spektrometer_sonya7III = (
measured_luminance_spektrometer_5000_1600 +
measured_luminance_spektrometer_5000_800 +
measured_luminance_spektrometer_5000_400 +
measured_luminance_spektrometer_5000_200 +
measured_luminance_spektrometer_5000 +
measured_luminance_spektrometer_2500 +
measured_luminance_spektrometer_1250 +
measured_luminance_spektrometer_625 +
measured_luminance_spektrometer_312 +
measured_luminance_spektrometer_156 +
measured_luminance_spektrometer_78 +
measured_luminance_spektrometer_39 +
measured_luminance_spektrometer_19)
212     full_well_capacity_a7III = [a / b for a, b in zip(
measured_luminance_spektrometer_sonya7III, sonya7III_mean)]
213     full_well_capacity_a7III = np.mean(np.array(
full_well_capacity_a7III))
214     plot_gammapcurve(sonya7III_mean,
measured_luminance_spektrometer_sonya7III,
full_well_capacity_a7III, 'Sony A7III', 'Export/Messung/
M05_Gammapcurve/Luminanz_SonyA7III.png')
215
216     # Panasonic Lumix GH5s
217     lumixgh5s_signal, lumixgh5s_mean = measure_gammapcurve(
image_directory_signal = 'Assets/Messung/M05_Gammapcurve/Lumix
GH5s')
218     measured_luminance_spektrometer_lumixgh5s = (
measured_luminance_spektrometer_5000_1600 +
measured_luminance_spektrometer_5000_800 +
measured_luminance_spektrometer_5000_400 +
measured_luminance_spektrometer_5000_200 +
measured_luminance_spektrometer_5000 +
measured_luminance_spektrometer_2500 +
measured_luminance_spektrometer_1250 +
measured_luminance_spektrometer_625 +
measured_luminance_spektrometer_312 +
measured_luminance_spektrometer_156 +
measured_luminance_spektrometer_78 +
measured_luminance_spektrometer_39 +
measured_luminance_spektrometer_19)
219     full_well_capacity_gh5s = [a / b for a, b in zip(
measured_luminance_spektrometer_lumixgh5s, lumixgh5s_mean)]
220     full_well_capacity_gh5s = np.mean(np.array(
```

```
full_well_capacity_gh5s))
221 plot_gammacurve(lumixgh5s_mean,
measured_luminance_spektrometer_lumixgh5s,
full_well_capacity_gh5s, 'Lumix GH5s', 'Export/Messung/
M05_Gammacurve/Luminanz_LumixGH5s.png')
222
223 #Phantom 4 Pro
224 phantom4_signal, phantom4_mean = measure_gammacurve(
image_directory_signal = 'Assets/Messung/M05_Gammacurve/Phantom
4 Pro')
225 measured_luminance_spektrometer_phantom4 = (
measured_luminance_spektrometer_5000_1600 +
measured_luminance_spektrometer_5000_800 +
measured_luminance_spektrometer_5000_400 +
measured_luminance_spektrometer_5000_200 +
measured_luminance_spektrometer_5000 +
measured_luminance_spektrometer_2500 +
measured_luminance_spektrometer_1250 +
measured_luminance_spektrometer_625 +
measured_luminance_spektrometer_312 +
measured_luminance_spektrometer_156 +
measured_luminance_spektrometer_78 +
measured_luminance_spektrometer_39 +
measured_luminance_spektrometer_19)
226 full_well_capacity_phantom4 = [a / b for a, b in zip(
measured_luminance_spektrometer_phantom4, phantom4_mean)]
227 full_well_capacity_phantom4 = np.mean(np.array(
full_well_capacity_phantom4))
228 plot_gammacurve(phantom4_mean,
measured_luminance_spektrometer_phantom4,
full_well_capacity_phantom4, 'Phantom 4 Pro', 'Export/Messung/
M05_Gammacurve/Luminanz_Phantom4.png')
229
230 #GoPro 10
231 gopro10_signal, gopro10_mean = measure_gammacurve(
image_directory_signal = 'Assets/Messung/M05_Gammacurve/GoPro/
DNG')
232 measured_luminance_spektrometer_gopro10 = (
measured_luminance_spektrometer_5000_800 +
measured_luminance_spektrometer_5000_400 +
measured_luminance_spektrometer_5000_200 +
measured_luminance_spektrometer_5000 +
measured_luminance_spektrometer_2500 +
measured_luminance_spektrometer_1250 +
measured_luminance_spektrometer_625 +
measured_luminance_spektrometer_312 +
measured_luminance_spektrometer_156 +
measured_luminance_spektrometer_78 +
measured_luminance_spektrometer_39 +
measured_luminance_spektrometer_19)
233 full_well_capacity_gopro10 = [a / b for a, b in zip(
measured_luminance_spektrometer_gopro10, gopro10_mean)]
234 full_well_capacity_gopro10 = np.mean(np.array(
full_well_capacity_gopro10))
235 plot_gammacurve(gopro10_mean,
measured_luminance_spektrometer_gopro10,
```

```
full_well_capacity_gopro10, 'GoPro 10', 'Export/Messung/
M05_Gammacurve/Luminanz_GoPro10.png')
236
237
238 plt.figure(figsize = (12, 6))
239
240 plt.plot(sonya7IV_mean,
measured_luminance_spektrometer_sonya7IV, marker = 'o',
linestyle='none', label = 'Sony A7IV')
241 plt.plot(sonya7III_mean,
measured_luminance_spektrometer_sonya7III, marker = 'o',
linestyle='none', label = 'Sony A7III')
242 plt.plot(lumixgh5s_mean,
measured_luminance_spektrometer_lumixgh5s, marker = 'o',
linestyle='none', label = 'Lumix GH5s')
243 plt.plot(phantom4_mean,
measured_luminance_spektrometer_phantom4, marker = 'o',
linestyle='none', label = 'Phantom 4 Pro')
244 plt.plot(gopro10_mean, measured_luminance_spektrometer_gopro10,
marker = 'o', linestyle='none', label = 'GoPro 10')
245
246 plt.axhline(y = 1, color = 'red', linestyle = '--', linewidth =
1, label = 'Messgrenze des Spektrometers')
247
248 plt.grid(which = 'both', linestyle = '--', linewidth = '0.5',
color = 'grey')
249
250 plt.xscale('log')
251 plt.yscale('log')
252
253 plt.xlabel('Signalwerte', labelpad = 20, fontsize = 12)
254 plt.ylabel('Luminanz in cd/m^2', labelpad = 20, fontsize = 12)
255
256 plt.tick_params(axis = 'both', which = 'major', labelsize = 11)
257 plt.tick_params(axis = 'both', which = 'minor', labelsize = 10)
258
259 plt.xlim(0, 2)
260
261 plt.legend(loc='upper left')
262
263 plt.savefig('Export/Messung/M05_Gammacurve/Luminanz_All.png',
transparent = False, dpi = 600)
264 plt.show()
265
266
```

Listing D.7: Gammakurve der Kameras

D.6. Quellcode Auswertung 6

```
1  import colour
2  import numpy as np
3  import pandas as pd
4
5
6  import matplotlib
7  matplotlib.use('TkAgg')
8  from matplotlib import pyplot as plt
9  from matplotlib.ticker import ScalarFormatter
10
11 from PIL import Image
12
13 import colormaps as cs
14
15 import os
16 import cv2
17
18 import rawpy
19 import imageio
20
21 import random
22
23 #plt.style.use('dark_background')
24
25 # Funktion zum einlesen von 8 bit Bildern
26 def read_image_8bit(file_path):
27     with rawpy.imread(file_path) as raw:
28         rgb_image = raw.postprocess(
29             use_camera_wb = True,
30             #no_auto_bright = True,
31             output_bps = 8
32         )
33     rgb_image = rgb_image / 256
34     return rgb_image
35
36 # Funktion zum einlesen von 16 bit Bildern
37 def read_image_16bit(file_path):
38
39     file_ending = file_path.split('.')[-1].lower()
40
41     if file_ending in ['nef', 'cr2', 'arw', 'dng', 'raw', 'rw2'
42 ]:
43         with rawpy.imread(file_path) as raw:
44             rgb_image = raw.postprocess(
45                 use_camera_wb = True,
46                 no_auto_bright = True,
47                 output_bps = 16,
48                 # output_color
49                 gamma = (1, 1)
50             )
51             rgb_image = rgb_image / 65535
52     elif file_ending in ['tif', 'tiff']:
```

```

53         rgb_image = Image.open(file_path)
54         rgb_image = np.array(rgb_image, dtype = np.float32)
55         rgb_image = rgb_image / 255
56
57     return rgb_image
58
59
60     def measure_vignette(img):
61         mean_brightness = []
62
63         height = img.shape[0]
64         width = img.shape[1]
65         center_height = height / 2
66         center_width = width / 2
67
68         # Berechnung der Entfernung jedes Pixels zum Mittelpunkt (
69         Satz des Pythagoras)
70         Y, X = np.ogrid[:height, :width]
71         distance_from_center = np.sqrt((X - center_width)**2 + (Y -
72         center_height)**2)
73
74         # Maximalen Abstand berechnen
75         max_distance = np.max(distance_from_center)
76
77         for distance in np.linspace(0, max_distance, 100):
78             mask = (distance_from_center >= distance - (
79             max_distance / 100)) & (distance_from_center < distance + (
80             max_distance / 100))
81             mean_brightness.append(np.mean(img[mask]))
82
83         return max_distance, mean_brightness
84
85     def plot_vignette(max_distance, mean_brightness, label:str,
86     link:str):
87         plt.figure(figsize = (12, 6))
88
89         plt.plot(np.linspace(0, max_distance, 100), mean_brightness
90         , marker = 'o', linestyle='--', label = label)
91
92         plt.grid(which = 'both', linestyle = '--', linewidth = '0.5
93         ', color = 'grey')
94
95         plt.xscale('linear')
96         plt.yscale('linear')
97
98         plt.xlabel('Relative Entfernung zur Bildmitte (Pixel)',
99         labelpad = 20, fontsize = 12)
100        plt.ylabel('Mittlerer Signalwert', labelpad = 20, fontsize
101        = 12)
102        plt.legend(loc='upper right')
103
104        plt.savefig(link, transparent = False, dpi = 600)
105        plt.show()

```

```
100     # Sony A7IV
101     image_a7IV = read_image_16bit('Assets/Messung/M06-
M10_Lumaeigenschaften/Sony A7IV/Signal/_DSC2480.ARW')
102     max_distance_a7IV, mean_brightness_a7IV = measure_vignette(
image_a7IV)
103     max_a7IV = max(mean_brightness_a7IV)
104     min_a7IV = min(mean_brightness_a7IV)
105     vignette_prozent_a7IV = (max_a7IV - min_a7IV)*100
106     plot_vignette(max_distance_a7IV, mean_brightness_a7IV, label =
'Sony A7IV', link = 'Export/Messung/M06-M10_Lumaeigenschaften/
Vignette/Vignette_Sony_A7IV.png')
107
108     # Sony A7III
109     image_a7III = read_image_16bit('Assets/Messung/M06-
M10_Lumaeigenschaften/Sony A7III/Signal/DSC01621.ARW')
110     max_distance_a7III, mean_brightness_a7III = measure_vignette(
image_a7III)
111     max_a7III = max(mean_brightness_a7III)
112     min_a7III = min(mean_brightness_a7III)
113     vignette_prozent_a7III = (max_a7III - min_a7III)*100
114     plot_vignette(max_distance_a7III, mean_brightness_a7III, label
= 'Sony A7III', link = 'Export/Messung/M06-M10_Lumaeigenschaften
/Vignette/Vignette_Sony_A7III.png')
115
116     # Panasonic Lumix GH5s
117     image_gh5s = read_image_16bit('Assets/Messung/M06-
M10_Lumaeigenschaften/Lumix GH5s/Signal/P1035090.RW2')
118     max_distance_gh5s, mean_brightness_gh5s = measure_vignette(
image_gh5s)
119     max_gh5s = max(mean_brightness_gh5s)
120     min_gh5s = min(mean_brightness_gh5s)
121     vignette_prozent_gh5s = (max_gh5s - min_gh5s)*100
122     plot_vignette(max_distance_gh5s, mean_brightness_gh5s, label =
'Panasonic Lumix GH5s', link = 'Export/Messung/M06-
M10_Lumaeigenschaften/Vignette/Vignette_Lumix_GH5s.png')
123
124     # Phantom 4 Pro
125     image_gh5s = read_image_16bit('Assets/Messung/M06-
M10_Lumaeigenschaften/Lumix GH5s/Signal/P1035090.RW2')
126     max_distance_gh5s, mean_brightness_gh5s = measure_vignette(
image_gh5s)
127     max_gh5s = max(mean_brightness_gh5s)
128     min_gh5s = min(mean_brightness_gh5s)
129     vignette_prozent_gh5s = (max_gh5s - min_gh5s)*100
130     plot_vignette(max_distance_gh5s, mean_brightness_gh5s, label =
'Panasonic Lumix GH5s', link = 'Export/Messung/M06-
M10_Lumaeigenschaften/Vignette/Vignette_Lumix_GH5s.png')
131
132     #GoPro 10
133     image_gopro = read_image_16bit('Assets/Messung/M06-
M10_Lumaeigenschaften/GoPro/Signal/DNG/GOPR0814.dng')
134     max_distance_gopro, mean_brightness_gopro = measure_vignette(
image_gopro)
135     max_gopro = max(mean_brightness_gopro)
136     min_gopro = min(mean_brightness_gopro)
137     vignette_prozent_gopro = (max_gopro - min_gopro)*100
```

```
138     plot_vignette(max_distance_gopro, mean_brightness_gopro, label  
139     = 'GoPro', link = 'Export/Messung/M06-M10_Lumaeigenschaften/  
140     Vignette/Vignette_GoPro.png')
```

Listing D.8: Vignettierung der Kameras

D.7. Quellcode Auswertung 7

```
1
2 import colour
3 import numpy as np
4 import pandas as pd
5
6
7 import matplotlib
8 matplotlib.use('TkAgg')
9 from matplotlib import pyplot as plt
10 from matplotlib.ticker import ScalarFormatter
11
12 from PIL import Image
13
14 import colorspacious as cs
15
16 import os
17 import cv2
18
19 import rawpy
20 import imageio
21
22 import random
23
24 #plt.style.use('dark_background')
25
26 # Funktion zum einlesen von 8 bit Bildern
27 def read_image_8bit(file_path):
28     with rawpy.imread(file_path) as raw:
29         rgb_image = raw.postprocess(
30             use_camera_wb = True,
31             #no_auto_bright = True,
32             output_bps = 8
33         )
34     rgb_image = rgb_image / 256
35     return rgb_image
36
37 # Funktion zum einlesen von 16 bit Bildern
38 def read_image_16bit(file_path):
39
40     file_ending = file_path.split('.')[-1].lower()
41
42     if file_ending in ['nef', 'cr2', 'arw', 'dng', 'raw', 'rw2'
43 ]:
44         with rawpy.imread(file_path) as raw:
45             rgb_image = raw.postprocess(
46                 use_camera_wb = True,
47                 no_auto_bright = True,
48                 output_bps = 16,
49                 # output_color
50                 gamma = (1, 1)
51             )
52             rgb_image = rgb_image / 65535
```

```

53     elif file_ending in ['.tif', '.tiff']:
54         rgb_image = Image.open(file_path)
55         rgb_image = np.array(rgb_image, dtype = np.float32)
56         rgb_image = rgb_image / 255
57
58     return rgb_image
59
60 def normalize_image(image):
61     return image / image.max()
62
63 def crop_image(image, x1, y1, x2, y2):
64     return image[int(y1) : int(y2), int(x1) : int(x2)]
65
66     def list_file_names(link: str):
67         all_files = os.listdir(link)
68         raw_files = [file for file in all_files]
69         raw_files.sort()
70
71     return raw_files
72
73 def crop_sides(img, crop_size):
74     height = img.shape[0]
75     width = img.shape[1]
76     top = crop_size
77     bottom = height - crop_size
78     left = crop_size
79     right = width - crop_size
80     cropped_img = img[top:bottom, left:right]
81     return cropped_img
82
83 def measure_dsnu(image_directory_dark: str):
84     dark_images = []
85
86     # Namen und Anzahl der Dunkelbilder ermitteln
87     file_list_dark = list_file_names(image_directory_dark)
88
89
90     # Dunkelbilder laden
91     for path in file_list_dark[0:25]:
92         img = read_image_16bit(f'{image_directory_dark}/{path}')
93
94         img = crop_sides(img, 500)
95         dark_images.append(img)
96
97     num_images_dark = len(dark_images)
98
99     middle_DSNU_v = []
100    mean_DSNU_v = []
101    max_DSNU_v = []
102    min_DSNU_v = []
103
104    middle_DSNU_h = []
105    mean_DSNU_h = []
106    max_DSNU_h = []
107    min_DSNU_h = []

```

```

108     for i in range(0, num_images_dark, 5):
109
110         images = []
111
112         images.append(dark_images[i][:, :, 1])
113         images.append(dark_images[i + 1][:, :, 1])
114         images.append(dark_images[i + 2][:, :, 1])
115         images.append(dark_images[i + 3][:, :, 1])
116         images.append(dark_images[i + 4][:, :, 1])
117
118         mean_matrix = np.mean(images, axis = 0)
119         std_devs_matrix = np.std(images, axis = 0)
120
121         height_center = mean_matrix.shape[1] // 2
122         width_center = mean_matrix.shape[0] // 2
123
124         middle_DSNU_v.append(mean_matrix[:, height_center])
125         mean_DSNU_v.append(np.mean(mean_matrix, axis = 1))
126         max_DSNU_v.append(np.max(mean_matrix, axis = 1))
127         min_DSNU_v.append(np.min(mean_matrix, axis = 1))
128
129         middle_DSNU_h.append(mean_matrix[width_center, :])
130         mean_DSNU_h.append(np.mean(mean_matrix, axis = 0))
131         max_DSNU_h.append(np.max(mean_matrix, axis = 0))
132         min_DSNU_h.append(np.min(mean_matrix, axis = 0))
133
134     return middle_DSNU_v, mean_DSNU_v, max_DSNU_v, min_DSNU_v,
middle_DSNU_h, mean_DSNU_h, max_DSNU_h, min_DSNU_h
135
136     def plot_DSNU(middle_DSNU, mean_DSNU, max_DSNU, min_DSNU, link,
horizontal: bool = False, vertical: bool = False):
137         plt.figure(figsize = (9, 9))
138
139         if horizontal == True:
140             plt.plot(np.arange(middle_DSNU.size), middle_DSNU,
label = 'Middle')
141             plt.plot(np.arange(mean_DSNU.size), mean_DSNU, label =
'Mean')
142             plt.plot(np.arange(max_DSNU.size), max_DSNU, label = '
Max')
143             plt.plot(np.arange(min_DSNU.size), min_DSNU, label = '
Min')
144
145             plt.xlabel('Pixelposition', labelpad = 20, fontsize =
12)
146             plt.ylabel('Signalwert (DU)', labelpad = 20, fontsize =
12)
147
148         if vertical == True:
149             plt.plot(middle_DSNU, np.arange(middle_DSNU.size),
label = 'Middle')
150             plt.plot(mean_DSNU, np.arange(mean_DSNU.size), label =
'Mean')
151             plt.plot(max_DSNU, np.arange(max_DSNU.size), label = '
Max')
152             plt.plot(min_DSNU, np.arange(min_DSNU.size), label = '

```

```
Min')
153
154     plt.xlabel('Signalwert (DU)', labelpad = 20, fontsize =
155     12)
156     plt.ylabel('Pixelposition', labelpad = 20, fontsize =
157     12)
158     plt.grid(which = 'both', linestyle = '--', linewidth = '0.5
159     ', color = 'grey')
160     plt.xscale('linear')
161     plt.yscale('linear')
162     plt.tick_params(axis = 'both', which = 'major', labelsiz
163     e = 11)
164     plt.tick_params(axis = 'both', which = 'minor', labelsiz
165     e = 10)
166     plt.legend(loc = 'upper right')
167     plt.savefig(link, transparent = False, dpi = 600)
168     plt.show()
169
170
171     # Sony A7IV
172     middle_DSNU_v_A7IV, mean_DSNU_v_A7IV, max_DSNU_v_A7IV,
173     min_DSNU_v_A7IV, middle_DSNU_h_A7IV, mean_DSNU_h_A7IV,
174     max_DSNU_h_A7IV, min_DSNU_h_A7IV = measure_dsnu('Assets/Messung/
175     M06-M10_Lumaeigenschaften/Sony A7IV/Dunkelbilder')
176     plot_DSNU(middle_DSNU_h_A7IV [3], mean_DSNU_h_A7IV [3],
177     max_DSNU_h_A7IV [3], min_DSNU_h_A7IV [3], link = 'Export/Messung/
178     M06-M10_Lumaeigenschaften/DSNU/DSNU_horizontal_A7IV.png',
179     horizontal = True)
180     plot_DSNU(middle_DSNU_v_A7IV [3], mean_DSNU_v_A7IV [3],
181     max_DSNU_v_A7IV [3], min_DSNU_v_A7IV [3], link = 'Export/Messung/
182     M06-M10_Lumaeigenschaften/DSNU/DSNU_vertical_A7IV.png', vertical
183     = True)
184
185     # Sony A7III
186     middle_DSNU_v_A7III, mean_DSNU_v_A7III, max_DSNU_v_A7III,
187     min_DSNU_v_A7III, middle_DSNU_h_A7III, mean_DSNU_h_A7III,
188     max_DSNU_h_A7III, min_DSNU_h_A7III = measure_dsnu('Assets/
189     Messung/M06-M10_Lumaeigenschaften/Sony A7III/Dunkelbilder')
190     plot_DSNU(middle_DSNU_h_A7III [3], mean_DSNU_h_A7III [3],
191     max_DSNU_h_A7III [3], min_DSNU_h_A7III [3], link = 'Export/Messung
192     /M06-M10_Lumaeigenschaften/DSNU/DSNU_horizontal_A7III.png',
193     horizontal = True)
194     plot_DSNU(middle_DSNU_v_A7III [3], mean_DSNU_v_A7III [3],
195     max_DSNU_v_A7III [3], min_DSNU_v_A7III [3], link = 'Export/Messung
196     /M06-M10_Lumaeigenschaften/DSNU/DSNU_vertical_A7III.png',
197     vertical = True)
198
199     # Panasonic Lumix GH5s
200     middle_DSNU_v_gh5s, mean_DSNU_v_gh5s, max_DSNU_v_gh5s,
201     min_DSNU_v_gh5s, middle_DSNU_h_gh5s, mean_DSNU_h_gh5s,
202     max_DSNU_h_gh5s, min_DSNU_h_gh5s = measure_dsnu('Assets/Messung/
```

```
M06-M10_Lumaeigenschaften/Lumix GH5s/Dunkelbilder')
183 plot_DSNU(middle_DSNU_h_gh5s[1], mean_DSNU_h_gh5s[1],
max_DSNU_h_gh5s[1], min_DSNU_h_gh5s[1], link = 'Export/Messung/
M06-M10_Lumaeigenschaften/DSNU/DSNU_horizontal_GH5s.png',
horizontal = True)
184 plot_DSNU(middle_DSNU_v_gh5s[1], mean_DSNU_v_gh5s[1],
max_DSNU_v_gh5s[1], min_DSNU_v_gh5s[1], link = 'Export/Messung/
M06-M10_Lumaeigenschaften/DSNU/DSNU_vertical_GH5s.png', vertical
= True)
185
186 # Phantom 4 Pro
187 middle_DSNU_v_phantom4, mean_DSNU_v_phantom4,
max_DSNU_v_phantom4, min_DSNU_v_phantom4, middle_DSNU_h_phantom4
, mean_DSNU_h_phantom4, max_DSNU_h_phantom4, min_DSNU_h_phantom4
= measure_dsnu('Assets/Messung/M06-M10_Lumaeigenschaften/
Phantom 4 Pro/Dunkelbilder')
188 plot_DSNU(middle_DSNU_h_phantom4[0], mean_DSNU_h_phantom4[0],
max_DSNU_h_phantom4[0], min_DSNU_h_phantom4[0], link = 'Export/
Messung/M06-M10_Lumaeigenschaften/DSNU/DSNU_horizontal_Phantom4.
png', horizontal = True)
189 plot_DSNU(middle_DSNU_v_phantom4[0], mean_DSNU_v_phantom4[0],
max_DSNU_v_phantom4[0], min_DSNU_v_phantom4[0], link = 'Export/
Messung/M06-M10_Lumaeigenschaften/DSNU/DSNU_vertical_Phantom4.
png', vertical = True)
190
191 # GoPro 10
192 middle_DSNU_v_gopro, mean_DSNU_v_gopro, max_DSNU_v_gopro,
min_DSNU_v_gopro, middle_DSNU_h_gopro, mean_DSNU_h_gopro,
max_DSNU_h_gopro, min_DSNU_h_gopro = measure_dsnu('Assets/
Messung/M06-M10_Lumaeigenschaften/GoPro/Dunkelbilder/DNG')
193 plot_DSNU(middle_DSNU_h_gopro[0], mean_DSNU_h_gopro[0],
max_DSNU_h_gopro[0], min_DSNU_h_gopro[0], link = 'Export/Messung
/M06-M10_Lumaeigenschaften/DSNU/DSNU_horizontal_GoPro.png',
horizontal = True)
194 plot_DSNU(middle_DSNU_v_gopro[0], mean_DSNU_v_gopro[0],
max_DSNU_v_gopro[0], min_DSNU_v_gopro[0], link = 'Export/Messung
/M06-M10_Lumaeigenschaften/DSNU/DSNU_vertical_GoPro.png',
vertical = True)
195
196
```

Listing D.9: DSNU der Kameras

D.8. Quellcode Auswertung 8

```
1     import colour
2     import numpy as np
3     import pandas as pd
4
5
6     import matplotlib
7     matplotlib.use('TkAgg')
8     from matplotlib import pyplot as plt
9     from matplotlib.ticker import ScalarFormatter
10
11    from PIL import Image
12
13    import colorspacious as cs
14
15    import os
16    import cv2
17
18    import rawpy
19    import imageio
20
21    import random
22
23    #plt.style.use('dark_background')
24
25    # Funktion zum einlesen von 8 bit Bildern
26    def read_image_8bit(file_path):
27        with rawpy.imread(file_path) as raw:
28            rgb_image = raw.postprocess(
29                use_camera_wb = True,
30                #no_auto_bright = True,
31                output_bps = 8
32            )
33        rgb_image = rgb_image / 256
34        return rgb_image
35
36    # Funktion zum einlesen von 16 bit Bildern
37    def read_image_16bit(file_path):
38
39        file_ending = file_path.split('.')[1].lower()
40
41        if file_ending in ['nef', 'cr2', 'arw', 'dng', 'raw', 'rw2'
42]:
43            with rawpy.imread(file_path) as raw:
44                rgb_image = raw.postprocess(
45                    use_camera_wb = True,
46                    no_auto_bright = True,
47                    output_bps = 16,
48                    # output_color
49                    gamma = (1, 1)
50                )
51            rgb_image = rgb_image / 65535
52        elif file_ending in ['tif', 'tiff']:
```

```

53         rgb_image = Image.open(file_path)
54         rgb_image = np.array(rgb_image, dtype = np.float32)
55         rgb_image = rgb_image / 255
56
57         return rgb_image
58
59     def normalize_image(image):
60         return image / image.max()
61
62     def crop_image(image, x1, y1, x2, y2):
63         return image[int(y1) : int(y2), int(x1) : int(x2)]
64
65     def list_file_names(link: str):
66         all_files = os.listdir(link)
67         raw_files = [file for file in all_files]
68         raw_files.sort()
69
70         return raw_files
71
72     def crop_sides(img, crop_size):
73         height = img.shape[0]
74         width = img.shape[1]
75         top = crop_size
76         bottom = height - crop_size
77         left = crop_size
78         right = width - crop_size
79         cropped_img = img[top:bottom, left:right]
80         return cropped_img
81
82     def measure_prnu(image_directory_signal: str,
83 image_directory_dark: str, i: int):
84         x = i * 5
85         y = (i * 5) + 5
86
87         dark_images = []
88         means_dark = []
89         black_level = []
90
91         signal_images = []
92
93         # Namen und Anzahl der Dunkelbilder ermitteln
94         file_list_dark = list_file_names(image_directory_dark)
95         file_list_dark = file_list_dark[x:y]
96         num_images_dark = len(file_list_dark)
97
98         # Namen und Anzahl der Signalbilder ermitteln
99         file_list_signal = list_file_names(image_directory_signal)
100        file_list_signal = file_list_signal[i]
101
102        # Dunkelbilder laden
103        for path in file_list_dark:
104            img = read_image_16bit(f'{image_directory_dark}/{path}',
105            )
106            img = crop_sides(img, 500)
107            dark_images.append(img)
108            means_dark.append(np.mean(img))

```

```

107
108     # Blacklevel und Read Noise berechnen
109     for i in range(0, num_images_dark, 5):
110         black_level.append((means_dark[i] + means_dark[i + 1] +
111                             means_dark[i + 2] + means_dark[i + 3] + means_dark[i + 4]) / 5)
112
113     # Signalbilder laden
114     img = read_image_16bit(f'{image_directory_signal}/{
115 file_list_signal}')
116     img = img - black_level[0]
117     img = crop_sides(img, 500)
118     signal_images.append(img)
119     mean_signal = np.mean(img)
120
121     deviation = (signal_images - mean_signal) / mean_signal *
122
123     return deviation
124
125 def plot_PRNU(deviation, label, link):
126     plt.figure(figsize=(12, 6))
127
128     plt.hist(deviation.flatten(), bins = 500, log = True, color
129 = 'red', alpha = 0.7, label = label)
130
131     plt.grid(which = 'both', linestyle = '--', linewidth = '0.5
132 ', color = 'grey')
133
134     plt.xscale('linear')
135     plt.yscale('log')
136
137     plt.xlabel('Abweichung (%)')
138     plt.ylabel('Anzahl der Pixel')
139     plt.legend(loc = 'upper left')
140
141     # Achsen als Ganzzahl formatieren
142     ax = plt.gca()
143     ax.xaxis.set_major_formatter(ScalarFormatter())
144     ax.yaxis.set_major_formatter(ScalarFormatter())
145
146     plt.savefig(link, transparent = False, dpi = 600)
147     plt.show()
148
149 # Sony A7IV
150 deviation_a7IV = measure_prnu(image_directory_dark = 'Assets/
151 Messungen/003_Luminance_Vignette/Sony A7IV/Dunkelbilder',
152 image_directory_signal = 'Assets/Messungen/003
153 _Luminance_Vignette/Sony A7IV/Signal', i = 3)
154 plot_PRNU(deviation_a7IV, label = 'Abweichung der Sony A7IV',
155 link = 'Export/Messungen/003_Luminance_Vignette/PRNU/PRNU_a7IV.
156 png')
157
158 # Sony A7III
159 deviation_a7III = measure_prnu(image_directory_dark = 'Assets/
160 Messungen/003_Luminance_Vignette/Sony A7III/Dunkelbilder',
161 image_directory_signal = 'Assets/Messungen/003

```

```
151     _Luminance_Vignette/Sony A7III/Signal', i = 3)
152     plot_PRNU(deviation_a7III, label = 'Abweichung der Sony A7III',
153             link = 'Export/Messungen/003_Luminance_Vignette/PRNU/PRNU_a7III
154             .png')
155
156     # Panasonic Lumix GH5s
157     deviation_gh5s = measure_prnu(image_directory_dark = 'Assets/
158     Messungen/003_Luminance_Vignette/Lumix GH5s/Dunkelbilder',
159     image_directory_signal = 'Assets/Messungen/003
160     _Luminance_Vignette/Lumix GH5s/Signal', i = 1)
161     plot_PRNU(deviation_gh5s, label = 'Abweichung der Panasonic
162     Lumix GH5s', link = 'Export/Messungen/003_Luminance_Vignette/
163     PRNU/PRNU_GH5s.png')
164
165     # Phantom 4 Pro
166     deviation_phantom4 = measure_prnu(image_directory_dark = '
167     Assets/Messungen/003_Luminance_Vignette/Phantom 4 Pro/
168     Dunkelbilder', image_directory_signal = 'Assets/Messungen/003
169     _Luminance_Vignette/Phantom 4 Pro/Signal', i = 0)
170     plot_PRNU(deviation_phantom4, label = 'Abweichung der Phantom 4
171     Pro', link = 'Export/Messungen/003_Luminance_Vignette/PRNU/
172     PRNU_phantom4.png')
173
174     # GoPro 10
175     deviation_gopro = measure_prnu(image_directory_dark = 'Assets/
176     Messungen/003_Luminance_Vignette/GoPro/Dunkelbilder/DNG',
177     image_directory_signal = 'Assets/Messungen/003
178     _Luminance_Vignette/GoPro/Signal/DNG', i = 0)
179     plot_PRNU(deviation_gopro, label = 'Abweichung der GoPro', link
180     = 'Export/Messungen/003_Luminance_Vignette/PRNU/PRNU_Gopro.png'
181     )
182
183
184
185
186
```

Listing D.10: PRNU der Kameras

D.9. Quellcode Auswertung 9, 10

```
1     import colour
2     import numpy as np
3     import pandas as pd
4
5
6     import matplotlib
7     matplotlib.use('TkAgg')
8     from matplotlib import pyplot as plt
9     from matplotlib.ticker import ScalarFormatter
10
11    from PIL import Image
12
13    import colorspacious as cs
14
15    import os
16    import cv2
17
18    import rawpy
19    import imageio
20
21    import random
22
23    #plt.style.use('dark_background')
24
25
26    def sRGB_to_Linear(x):
27        result = np.where(x <= 0.04045, (x / 12.92), np.power(((x +
28        0.055) / 1.055), 2.4) )
29        return result
30
31    def Linear_to_sRGB(x):
32        result = np.where(x <= 0.0031308, (x * 12.92), 1.055 * np.
33        power(x, (1 / 2.4)) - 0.055 )
34        return result
35
36    # Funktion zum einlesen von 8 bit Bildern
37    def read_image_8bit(file_path):
38        with rawpy.imread(file_path) as raw:
39            rgb_image = raw.postprocess(
40                use_camera_wb = True,
41                #no_auto_bright = True,
42                output_bps = 8
43            )
44            rgb_image = rgb_image / 256
45            return rgb_image
46
47    # Funktion zum einlesen von 16 bit Bildern
48    def read_image_16bit(file_path):
49
50        file_ending = file_path.split('.')[0].lower()
51
52        if file_ending in ['nef', 'cr2', 'arw', 'dng', 'raw', 'rw2']:
```

```
51         with rawpy.imread(file_path) as raw:
52             rgb_image = raw.postprocess(
53                 use_camera_wb = True,
54                 no_auto_bright = True,
55                 output_bps = 16,
56                 # output_color
57                 gamma = (1, 1)
58             )
59         rgb_image = rgb_image / 65535
60
61     elif file_ending in ['.tif', '.tiff']:
62         rgb_image = Image.open(file_path)
63         rgb_image = np.array(rgb_image, dtype = np.float32)
64         rgb_image = rgb_image / 255
65
66     return rgb_image
67
68 def normalize_image(image):
69     return image / image.max()
70
71 def crop_image(image, x1, y1, x2, y2):
72     return image[int(y1) : int(y2), int(x1) : int(x2)]
73
74 def list_file_names(link: str):
75     all_files = os.listdir(link)
76     raw_files = [file for file in all_files]
77     raw_files.sort()
78
79     return raw_files
80
81 def crop_sides(img, crop_size):
82     height = img.shape[0]
83     width = img.shape[1]
84     top = crop_size
85     bottom = height - crop_size
86     left = crop_size
87     right = width - crop_size
88     cropped_img = img[top:bottom, left:right]
89     return cropped_img
90
91 def measure_snr(image_directory_dark: str,
92 image_directory_signal: str):
93     dark_images = []
94     means_dark = []
95     std_devs_dark = []
96
97     black_level = []
98     read_noise = []
99
100    signal_images = []
101    means_signal = []
102    std_devs_signal = []
103
104    # Namen und Anzahl der Dunkelbilder ermitteln
105    file_list_dark = list_file_names(image_directory_dark)
```

```

106         num_images_dark = len(file_list_dark)
107
108
109         # Dunkelbilder laden
110         for path in file_list_dark:
111             img = read_image_16bit(f'{image_directory_dark}/{path}'
)
112             img = crop_sides(img, 500)
113             means_dark.append(np.mean(img))
114             std_devs_dark.append(np.std(img))
115
116
117         # Blacklevel und Read Noise berechnen
118         for i in range(0, num_images_dark, 5):
119             black_level.append((means_dark[i] + means_dark[i + 1] +
means_dark[i + 2] + means_dark[i + 3] + means_dark[i + 4]) / 5)
120             read_noise.append((std_devs_dark[i] + std_devs_dark[i +
1] + std_devs_dark[i + 2] + std_devs_dark[i + 3] +
std_devs_dark[i + 4]) / 5)
121
122
123         # Namen und Anzahl der Dunkelbilder ermitteln
124         file_list_signal = list_file_names(image_directory_signal)
125         num_images_signal = len(file_list_signal)
126
127
128         # Signalbilder laden
129         c = 0
130
131         for path in file_list_signal:
132             img = read_image_16bit(f'{image_directory_signal}/{path
}')
133             img = img - black_level[c]
134             img = crop_sides(img, 500)
135             means_signal.append(np.mean(img))
136             std_devs_signal.append(np.std(img))
137             c += 1
138
139
140         # SNRs berechnen
141         # Quelle: https://www.imatest.com/support/docs/23-1/dynamic
/
142         snr = np.array(means_signal) / np.array(read_noise)
143         snr_db = 20 * np.log10(snr)
144         snr_density_units = 10 * np.log10(snr)
145         snr_f_stops = np.log2(snr)
146
147         calculated_full_well_snr = (np.array(means_signal) / np.max
(means_signal)) / np.array(read_noise)
148         calculated_snr_db = 20 * np.log10(calculated_full_well_snr)
149         calculated_snr_density_units = 10 * np.log10(
calculated_full_well_snr)
150         calculated_snr_f_stops = np.log2(calculated_full_well_snr)
151
152         # Ausgabe
153         return dark_images, signal_images, black_level, read_noise,

```

```
means_signal, std_devs_signal, snr_db, snr_density_units,
snr_f_stops, num_images_signal, num_images_dark,
calculated_full_well_snr, calculated_snr_db,
calculated_snr_density_units, calculated_snr_f_stops
154
155 def plot_read_noise(iso, read_noise, label: str, link):
156     plt.figure(figsize = (12, 6))
157
158     plt.plot(iso, read_noise, marker = 'o', linestyle='-',
label = label)
159
160     plt.grid(which = 'both', linestyle = '--', linewidth = '0.5
', color = 'grey')
161
162     plt.xscale('log')
163     plt.yscale('log')
164
165     plt.xlabel('Sensor Vorverstaerkung (ISO)')
166     plt.ylabel('Read Noise (DU)')
167     plt.legend(loc='upper left')
168
169     # Achsen als Ganzzahl formatieren
170     ax = plt.gca()
171     ax.xaxis.set_major_formatter(ScalarFormatter())
172     ax.yaxis.set_major_formatter(ScalarFormatter())
173
174     plt.savefig(link, transparent = False, dpi = 600)
175     plt.show()
176
177 def plot_snr(iso, snr, calculated_full_well_snr, label: str,
link):
178     plt.figure(figsize = (12, 6))
179
180     plt.plot(iso, snr, marker = 'o', linestyle='-', label =
label)
181     plt.plot(iso, calculated_full_well_snr, marker = 'o',
linestyle='-', label = 'errechneter SNR bei Vollaussteuerung des
Sensors')
182
183     plt.grid(which = 'both', linestyle = '--', linewidth = '0.5
', color = 'grey')
184
185     plt.xscale('log')
186     plt.yscale('log')
187
188     plt.xlabel('Sensor Vorverstaerkung (ISO)')
189     plt.ylabel('SNR (f-stops)')
190     plt.legend(loc = 'lower left')
191
192     # Achsen als Ganzzahl formatieren
193     ax = plt.gca()
194     ax.xaxis.set_major_formatter(ScalarFormatter())
195     ax.yaxis.set_major_formatter(ScalarFormatter())
196
197     plt.savefig(link, transparent = False, dpi = 600)
198     plt.show()
```



```
19.53125]
229  luminance_spectrometer_a7IV = [2721, 2142, 1729, 1396, 1132,
888.8, 715.4, 568.2, 452.7, 362.9, 291, 227.3, 181.8, 144.7,
114.6, 88.14, 69.85, 55, 42.21, 32.88, 25.57, 19.33, 14.94,
11.86, 9.767, 9.767, 9.767, 9.767, 9.767, 9.767, 9.767, 9.767,
9.767, 9.767, 9.767, 9.767, 9.767]
230
231  plot_read_noise(iso = iso_a7IV, read_noise = read_noise_a7IV,
label = 'Sony A7IV', link = 'Export/Messung/M06-
M10_Lumaeigenschaften/Noise/Noise_SonyA7IV.png')
232  plot_snr(iso = iso_a7IV, snr = snr_f_stops_a7IV,
calculated_full_well_snr = calculated_snr_f_stops_a7IV, label =
'Sony A7IV', link = 'Export/Messung/M06-M10_Lumaeigenschaften/
SNR/SNR_SonyA7IV.png')
233  plot_luminanz_signal(mean_signal = means_signal_a7IV,
luminance_spectrometer = luminance_spectrometer_a7IV, label = '
Sony A7IV', link = 'Export/Messung/M06-M10_Lumaeigenschaften/
Luminance_Signal/Luminance_Signal_SonyA7IV.png')
234
235
236  # Sony A7III
237  dark_images_a7III, signal_images_a7III, black_level_a7III,
read_noise_a7III, means_signal_a7III, std_devs_a7III,
snr_db_a7III, snr_density_units_a7III, snr_f_stops_a7III,
num_images_signal_a7III, num_images_dark_a7III,
calculated_full_well_snr_a7III, calculated_snr_db_a7III,
calculated_snr_density_units_a7III, calculated_snr_f_stops_a7III
= measure_snr(image_directory_signal = 'Assets/Messung/M06-
M10_Lumaeigenschaften/Sony A7III/Signal', image_directory_dark =
'Assets/Messung/M06-M10_Lumaeigenschaften/Sony A7III/
Dunkelbilder')
238  iso_a7III = [50, 64, 80, 100, 125, 160, 200, 250, 320, 400,
500, 640, 800, 1000, 1250, 1600, 2000, 2500, 3200, 4000, 5000,
6400, 8000, 10000, 12800, 16000, 20000, 25600, 32000, 40000,
51200, 64000, 80000, 102400, 128000, 160000, 204800]
239  luminance_luxmeter_a7III = [5000, 3906.25, 3125, 2500, 2000,
1562.5, 1250, 1000, 781.25, 625, 500, 390.625, 312.5, 250, 200,
156.25, 125, 100, 78.125, 62.5, 50, 39.0625, 31.25, 25,
19.53125, 19.53125, 19.53125, 19.53125, 19.53125, 19.53125,
19.53125, 19.53125, 19.53125, 19.53125, 19.53125, 19.53125,
19.53125]
240  luminance_spectrometer_a7III = [2721, 2142, 1729, 1396, 1132,
888.8, 715.4, 568.2, 452.7, 362.9, 291, 227.3, 181.8, 144.7,
114.6, 88.14, 69.85, 55, 42.21, 32.88, 25.57, 19.33, 14.94,
11.86, 9.767, 9.767, 9.767, 9.767, 9.767, 9.767, 9.767, 9.767,
9.767, 9.767, 9.767, 9.767, 9.767]
241
242  plot_read_noise(iso = iso_a7III, read_noise = read_noise_a7III,
label = 'Sony A7III', link = 'Export/Messung/M06-
M10_Lumaeigenschaften/Noise/Noise_SonyA7III.png')
243  plot_snr(iso = iso_a7III, snr = snr_f_stops_a7III,
calculated_full_well_snr = calculated_snr_f_stops_a7III, label =
'Sony A7III', link = 'Export/Messung/M06-M10_Lumaeigenschaften/
SNR/SNR_SonyA7III.png')
244  plot_luminanz_signal(mean_signal = means_signal_a7III,
luminance_spectrometer = luminance_spectrometer_a7III, label = '
```

```
Sony A7III', link = 'Export/Messung/M06-M10_Lumaeigenschaften/  
Luminance_Signal/Luminance_Signal_SonyA7III.png')  
245  
246  
247 # Panasonic Lumix GH5s  
248 dark_images_gh5s, signal_images_gh5s, black_level_gh5s,  
read_noise_gh5s, means_signal_gh5s, std_devs_gh5s, snr_db_gh5s,  
snr_density_units_gh5s, snr_f_stops_gh5s, num_images_signal_gh5s  
, num_images_dark_gh5s, calculated_full_well_snr_gh5s,  
calculated_snr_db_gh5s, calculated_snr_density_units_gh5s,  
calculated_snr_f_stops_gh5s = measure_snr(image_directory_signal  
= 'Assets/Messung/M06-M10_Lumaeigenschaften/Lumix GH5s/Signal',  
image_directory_dark = 'Assets/Messung/M06-  
M10_Lumaeigenschaften/Lumix GH5s/Dunkelbilder')  
249 iso_gh5s = [80, 100, 125, 160, 200, 250, 320, 400, 500, 640,  
800, 1000, 1250, 1600, 2000, 2500, 3200, 4000, 5000, 6400, 8000,  
10000, 12800, 16000, 20000, 25600, 32000, 40000, 51200, 64000,  
80000, 102400, 128000, 160000, 204800]  
250 luminance_luxmeter_gh5s = [3125, 2500, 2000, 1562.5, 1250,  
1000, 781.25, 625, 500, 390.625, 312.5, 250, 200, 156.25, 125,  
100, 78.125, 62.5, 50, 39.0625, 31.25, 25, 19.53125, 19.53125,  
19.53125, 19.53125, 19.53125, 19.53125, 19.53125, 19.53125,  
19.53125, 19.53125, 19.53125, 19.53125, 19.53125]  
251 luminance_spectrometer_gh5s = [1729, 1396, 1132, 888.8, 715.4,  
568.2, 452.7, 362.9, 291, 227.3, 181.8, 144.7, 114.6, 88.14,  
69.85, 55, 42.21, 32.88, 25.57, 19.33, 14.94, 11.86, 9.767,  
9.767, 9.767, 9.767, 9.767, 9.767, 9.767, 9.767, 9.767,  
9.767, 9.767, 9.767]  
252  
253 plot_read_noise(iso = iso_gh5s, read_noise = read_noise_gh5s,  
label = 'Panasonic Lumix GH5s', link = 'Export/Messung/M06-  
M10_Lumaeigenschaften/Noise/Noise_LumixGH5s.png')  
254 plot_snr(iso = iso_gh5s, snr = snr_f_stops_gh5s,  
calculated_full_well_snr = calculated_snr_f_stops_gh5s, label =  
'Panasonic Lumix GH5s', link = 'Export/Messung/M06-  
M10_Lumaeigenschaften/SNR/SNR_LumixGH5s.png')  
255 plot_luminanz_signal(mean_signal = means_signal_gh5s,  
luminance_spectrometer = luminance_spectrometer_gh5s, label = '  
Panasonic Lumix GH5s', link = 'Export/Messung/M06-  
M10_Lumaeigenschaften/Luminance_Signal/  
Luminance_Signal_LumixGH5s.png')  
256  
257  
258 # Phantom 4 Pro  
259 dark_images_phantom4, signal_images_phantom4,  
black_level_phantom4, read_noise_phantom4, means_signal_phantom4  
, std_devs_phantom4, snr_db_phantom4, snr_density_units_phantom4  
, snr_f_stops_phantom4, num_images_signal_phantom4,  
num_images_dark_phantom4, calculated_full_well_snr_phantom4,  
calculated_snr_db_phantom4,  
calculated_snr_density_units_phantom4,  
calculated_snr_f_stops_phantom4 = measure_snr(  
image_directory_signal = 'Assets/Messung/M06-  
M10_Lumaeigenschaften/Phantom 4 Pro/Signal',  
image_directory_dark = 'Assets/Messung/M06-M10_Lumaeigenschaften  
/Phantom 4 Pro/Dunkelbilder')
```

```
260     iso_phantom4 = [100, 200, 400, 800, 1600, 3200, 6400, 12800]
261     luminance_luxmeter_phantom4 = [2500, 1250, 625, 312.5, 156.25,
262     78.125, 39.0625, 19.53125]
263     luminance_spectrometer_phantom4 = [1396, 715.4, 362.9, 181.8,
264     88.14, 42.21, 19.33, 9.767]
265
266     plot_read_noise(iso = iso_phantom4, read_noise =
267     read_noise_phantom4, label = 'Phantom 4 Pro', link = 'Export/
268     Messung/M06-M10_Lumaeigenschaften/Noise/Noise_Phantom4Pro.png')
269     plot_snr(iso = iso_phantom4, snr = snr_f_stops_phantom4,
270     calculated_full_well_snr = calculated_snr_f_stops_phantom4,
271     label = 'Phantom 4 Pro', link = 'Export/Messung/M06-
272     M10_Lumaeigenschaften/SNR/SNR_Phantom4Pro.png')
273     plot_luminanz_signal(mean_signal = means_signal_phantom4,
274     luminance_spectrometer = luminance_spectrometer_phantom4, label =
275     'Phantom 4 Pro', link = 'Export/Messung/M06-
276     M10_Lumaeigenschaften/Luminance_Signal/
277     Luminance_Signal_Phantom4Pro.png')
278
279     # GoPro 10
280     dark_images_gopro, signal_images_gopro, black_level_gopro,
281     read_noise_gopro, means_signal_gopro, std_devs_gopro,
282     snr_db_gopro, snr_density_units_gopro, snr_f_stops_gopro,
283     num_images_signal_gopro, num_images_dark_gopro,
284     calculated_full_well_snr_gopro, calculated_snr_db_gopro,
285     calculated_snr_density_units_gopro, calculated_snr_f_stops_gopro
286     = measure_snr(image_directory_signal = 'Assets/Messung/M06-
287     M10_Lumaeigenschaften/GoPro/Signal/DNG', image_directory_dark =
288     'Assets/Messung/M06-M10_Lumaeigenschaften/GoPro/Dunkelbilder/DNG
289     ')
290     iso_gopro = [100, 200, 400, 800, 1600, 3200]
291     luminance_luxmeter_gopro = [1250, 625, 312.5, 156.25, 78.125,
292     39.0625]
293     luminance_spectrometer_gopro = [715.4, 362.9, 181.8, 88.14,
294     42.21, 19.33]
295
296     plot_read_noise(iso = iso_gopro, read_noise = read_noise_gopro,
297     label = 'GoPro 10', link = 'Export/Messung/M06-
298     M10_Lumaeigenschaften/Noise/Noise_GoPro10.png')
299     plot_snr(iso = iso_gopro, snr = snr_f_stops_gopro,
300     calculated_full_well_snr = calculated_snr_f_stops_gopro, label =
301     'GoPro 10', link = 'Export/Messung/M06-M10_Lumaeigenschaften/
302     SNR/SNR_GoPro10.png')
303     plot_luminanz_signal(mean_signal = means_signal_gopro,
304     luminance_spectrometer = luminance_spectrometer_gopro, label = '
305     GoPro 10', link = 'Export/Messung/M06-M10_Lumaeigenschaften/
306     Luminance_Signal/Luminance_Signal_GoPro10.png')
```

Listing D.11: Rauschen und SNR der Kameras

D.10. Quellcode Auswertung 11

```
1     import colour
2     import numpy as np
3     import pandas as pd
4
5
6     import matplotlib
7     matplotlib.use('TkAgg')
8     from matplotlib import pyplot as plt
9     from matplotlib.ticker import ScalarFormatter
10
11    from PIL import Image
12
13    import colorspacious as cs
14
15    import os
16    import cv2
17
18    import rawpy
19    import imageio
20
21    import random
22
23    #plt.style.use('dark_background')
24
25    # Funktion zum einlesen von 8 bit Bildern
26    def read_image_8bit(file_path):
27        with rawpy.imread(file_path) as raw:
28            rgb_image = raw.postprocess(
29                use_camera_wb = True,
30                #no_auto_bright = True,
31                output_bps = 8
32            )
33        rgb_image = rgb_image / 256
34        return rgb_image
35
36    # Funktion zum einlesen von 16 bit Bildern
37    def read_image_16bit(file_path):
38
39        file_ending = file_path.split('.')[0].lower()
40
41        if file_ending in ['nef', 'cr2', 'arw', 'dng', 'raw', 'rw2'
42]:
43            with rawpy.imread(file_path) as raw:
44                rgb_image = raw.postprocess(
45                    use_camera_wb = True,
46                    no_auto_bright = True,
47                    output_bps = 16,
48                    # output_color
49                    gamma = (1, 1)
50                )
51                rgb_image = rgb_image / 65535
52        elif file_ending in ['tif', 'tiff']:
```

```
53         rgb_image = Image.open(file_path)
54         rgb_image = np.array(rgb_image, dtype = np.float32)
55         rgb_image = rgb_image / 255
56
57         return rgb_image
58
59     def normalize_image(image):
60         return image / image.max()
61
62     def crop_image(image, x1, y1, x2, y2):
63         return image[int(y1) : int(y2), int(x1) : int(x2)]
64
65     def list_file_names(link: str):
66         all_files = os.listdir(link)
67         raw_files = [file for file in all_files]
68         raw_files.sort()
69
70         return raw_files
71
72     def crop_sides(img, crop_size):
73         height = img.shape[0]
74         width = img.shape[1]
75         top = crop_size
76         bottom = height - crop_size
77         left = crop_size
78         right = width - crop_size
79         cropped_img = img[top:bottom, left:right]
80         return cropped_img
81
82     def measure_xy_positions_color(img):
83         plt.imshow(img, cmap = 'gray')
84         plt.title('Klicken Sie auf die Eckpunkte der Patches, um
85         einen Bereich auszuwaehlen')
86         points = plt.ginput(n = 48, timeout = 1000)
87         plt.close()
88
89         return points
90
91     def measure_color_patches(img):
92         patches = []
93         mean_patches = []
94
95         points = measure_xy_positions_color(img)
96
97         for i in range(0, 48, 2):
98             x1, y1 = points[i]
99             x2, y2 = points[i + 1]
100
101             cropped_image = crop_image(img, x1, y1, x2, y2)
102             patches.append(cropped_image)
103
104             mean_r = np.mean(cropped_image[:, :, 0])
105             mean_g = np.mean(cropped_image[:, :, 1])
106             mean_b = np.mean(cropped_image[:, :, 2])
107
108             mean_patches.append((mean_r, mean_g, mean_b))
```

```

108
109     return patches, mean_patches
110
111
112     reference_color_patches = pd.DataFrame(
113         {
114             "Color": ['Dark Skin', 'Light Skin', 'Blue Sky', '
Foliage', 'Blue Flower', 'Bluish Green', 'Orange', 'Purplish
blue', 'moderate red', 'purple', 'yellow green', 'orange yellow'
, 'blue flower', 'green', 'red', 'yellow', 'magenta', 'cyan', '
White', 'Neutral', 'Neutral', 'Neutral', 'Neutral', 'Black'],
115             "X": [14.58, 54.4, 15.34, 7.14, 49.31, 41.71, 47.89,
18.23, 40.32, 13.53, 60.19, 53.34, 8.6, 9.97, 37.11, 55.78,
43.54, 47.09, 95.05, 68.99, 41.54, 22.53, 11.07, 3.12],
116             "Y": [12.51, 49.35, 17.61, 12.41, 45.51, 57.74, 37.39,
18.27, 28.32, 8.25, 82.73, 45.59, 3.75, 23.31, 19.08, 63.14,
21.44, 59.26, 100, 72.74, 43.83, 23.77, 11.68, 3.3],
117             "Z": [8.65, 37.85, 30.46, 3.2, 85.69, 67.46, 6.82,
64.25, 18.75, 38.88, 33.24, 8.43, 45.32, 6.4, 2.55, 5.06, 52.53,
102.19, 108.88, 78.32, 46.9, 25.4, 12.37, 3.48],
118             "x": [0.41, 0.38, 0.24, 0.31, 0.27, 0.25, 0.52, 0.18,
0.46, 0.22, 0.34, 0.5, 0.15, 0.25, 0.63, 0.45, 0.37, 0.23, 0.31,
0.31, 0.31, 0.31, 0.32, 0.32],
119             "y": [0.35, 0.35, 0.28, 0.55, 0.25, 0.35, 0.41, 0.18,
0.32, 0.14, 0.47, 0.42, 0.07, 0.59, 0.32, 0.51, 0.18, 0.28,
0.33, 0.33, 0.33, 0.33, 0.33, 0.33]
120         }
121     )
122
123     xyz_reference = reference_color_patches[['X', 'Y', 'Z']] / 100
124
125     # White Point Correction
126     illuminant_D65 = np.array([0.31382, 0.33100])
127     illuminant_measurement = colour.models.XYZ_to_xy(xyz_reference.
loc[18])
128
129     reference_color_patches_d65 = colour.models.XYZ_to_RGB(
xyz_reference, "ITU-R BT.2020", illuminant =
illuminant_measurement)
130     xyz_reference_d65 = colour.models.RGB_to_XYZ(
reference_color_patches_d65, "ITU-R BT.2020", illuminant =
illuminant_D65)
131
132     reference_scene_linear_rgb = colour.models.XYZ_to_RGB(
xyz_reference_d65, "ITU-R BT.2020")
133     reference_signal_rgb = colour.models.oetf_BT2100_HLG(
reference_scene_linear_rgb)
134     reference_display_linear_rgb = colour.models.eotf_BT2100_HLG(
reference_signal_rgb)
135
136     reference_display_linear_XYZ = colour.models.RGB_to_XYZ(
reference_display_linear_rgb, "ITU-R BT.2020") / 1000
137     reference_display_linear_xy = colour.models.XYZ_to_xy(
reference_display_linear_XYZ)
138     reference_display_linear_Lab = colour.models.XYZ_to_Lab(
reference_display_linear_XYZ)

```

```
139
140
141     measured_color_patches = pd.read_csv('Assets/Messung/
M04_Verifikation_TE188/M04_Verifikation_TE188.csv', sep = ';')
142     xyz_measured = measured_color_patches[['X', 'Y', 'Z']] /
measured_color_patches['Y'][18]
143
144     # White Point Correction
145     illuminant_D65 = np.array([0.3127, 0.3290])
146     illuminant_measurement = colour.models.XYZ_to_xy(xyz_measured.
loc[18])
147
148     measured_color_patches_d65 = colour.models.XYZ_to_RGB(
xyz_measured, "ITU-R BT.2020", illuminant =
illuminant_measurement)
149     xyz_measured_d65 = colour.models.RGB_to_XYZ(
measured_color_patches_d65, "ITU-R BT.2020", illuminant =
illuminant_D65)
150
151     measured_scene_linear_rgb = colour.models.XYZ_to_RGB(
xyz_measured_d65, "ITU-R BT.2020")
152     measured_signal_rgb = colour.models.oetf_BT2100_HLG(
measured_scene_linear_rgb)
153     measured_display_linear_rgb = colour.models.eotf_BT2100_HLG(
measured_signal_rgb)
154
155     measured_display_linear_XYZ = colour.models.RGB_to_XYZ(
measured_display_linear_rgb, "ITU-R BT.2020") / 1000
156     measured_display_linear_xy = colour.models.XYZ_to_xy(
measured_display_linear_XYZ)
157     measured_display_linear_Lab = colour.models.XYZ_to_Lab(
measured_display_linear_XYZ)
158
159
160     # Panasonic UC4000
161     img_uc4000 = read_image_16bit('Assets/Messung/
M11_Spektraleigenschaften/Panasonic UC4000/TIFF/
NINJAV_S001_S001_T060_frame_1.tiff')
162     patches_uc4000, mean_uc4000 = measure_color_patches(img_uc4000)
163     mean_uc4000 = np.array([[ 0.50589518,  0.32668948,
0.35542297], [ 0.85037358,  0.76292626,  0.74654789], [
0.57521789,  0.64152089,  0.771222  ], [ 0.40502001,
0.59249808,  0.36376365], [ 0.66892639,  0.6127369 ,
0.78044913], [ 0.74794926,  0.89520667,  0.89699173], [
0.81885785,  0.58021298,  0.35013882], [ 0.31695414,
0.45530783,  0.71179547], [ 0.82468122,  0.55611339,
0.50071601], [ 0.47619882,  0.28381757,  0.71723133], [
0.76443725,  0.85553886,  0.7099967 ], [ 0.8113995 ,
0.63216074,  0.34268117], [ 0.15435491,  0.26766315,
0.75644437], [ 0.51748553,  0.7234846 ,  0.54311633], [
0.82675182,  0.16595813,  0.11978335], [ 0.86056313,
0.88614174,  0.35016116], [ 0.73024805,  0.2545387 ,  0.7151466
], [ 0.60434278,  0.77026794,  0.82315191], [ 0.96479178,
0.98164842,  1.          ], [ 0.86635774,  0.88388194,
0.90222343], [ 0.7355206 ,  0.75302719,  0.77128862], [
0.5671341 ,  0.58712716,  0.60144133], [ 0.38401872,
```

```

0.39771769, 0.40926218], [ 0.20175297, 0.2061647 ,
0.21624994]])
164 uc4000_signal_rgb = colour.models.eotf_BT2100_HLG(mean_uc4000 ,
0, 1000)
165 uc4000_XYZ_D65 = colour.models.RGB_to_XYZ(uc4000_signal_rgb, "
ITU-R BT.2020", illuminant = illuminant_D65)
166
167 uc4000_XYZ = pd.DataFrame(uc4000_XYZ_D65, columns = ['X', 'Y',
'Z'])
168 uc4000_XYZ = uc4000_XYZ / uc4000_XYZ['Y'][18]
169
170 uc4000_xy = colour.models.XYZ_to_xy(uc4000_XYZ)
171 uc4000_Lab = colour.models.XYZ_to_Lab(uc4000_XYZ)
172
173
174 # Ikegami UHK-X700
175 img_uhcx700 = read_image_16bit('Assets/Messung/
M11_Spektraleigenschaften/Ikegami UHK-X700/TIFF/
NINJAV_S001_S001_T175_frame_1.tiff')
176 patches_uhcx700, mean_uhcx700 = measure_color_patches(
img_uhcx700)
177 mean_uhcx700 = np.array([(0.34904882, 0.21897195, 0.20913641),
(0.64297646, 0.5370419, 0.51966912), (0.39422527, 0.43210906,
0.53165388), (0.25683054, 0.41018721, 0.19615583), (0.47850838,
0.41625687, 0.52921039), (0.51766157, 0.67679727, 0.65973514),
(0.61139947, 0.38403067, 0.23531471), (0.21134469, 0.30307075,
0.49124712), (0.61359674, 0.31940627, 0.35430557), (0.32933819,
0.17704082, 0.4685671), (0.55380255, 0.64604735, 0.44453606),
(0.60913837, 0.43478188, 0.22994581), (0.10065107, 0.1259958,
0.5421049), (0.32575661, 0.54428357, 0.28530511), (0.60539198,
0.06501776, 0.10694037), (0.65287274, 0.66507697, 0.2123201),
(0.53428966, 0.10641205, 0.44580403), (0.40606269, 0.56472504,
0.60253537), (0.74954104, 0.75485027, 0.74691981), (0.65947354,
0.66474843, 0.6555596), (0.53771734, 0.54214084, 0.5318867),
(0.39557227, 0.39649123, 0.38628733), (0.27082044, 0.27055946,
0.26188245), (0.14461279, 0.14456159, 0.13659349)])
178 uhcx700_signal_rgb = colour.models.eotf_BT2100_HLG(mean_uhcx700
, 0, 1000)
179 uhcx700_XYZ_D65 = colour.models.RGB_to_XYZ(uhcx700_signal_rgb ,
"ITU-R BT.2020", illuminant = illuminant_D65)
180 uhcx700_XYZ = pd.DataFrame(uhcx700_XYZ_D65, columns = ['X', 'Y',
'Z'])
181 uhcx700_XYZ = uhcx700_XYZ / uhcx700_XYZ['Y'][18]
182 uhcx700_xy = colour.models.XYZ_to_xy(uhcx700_XYZ)
183 uhcx700_Lab = colour.models.XYZ_to_Lab(uhcx700_XYZ)
184
185
186 colour.delta_E(measured_display_linear_Lab ,
reference_display_linear_Lab)
187 colour.delta_E(measured_display_linear_Lab , uc4000_Lab)
188 colour.delta_E(measured_display_linear_Lab , uhcx700_Lab)
189 colour.delta_E(uc4000_Lab , uhcx700_Lab)
190
191
192 fs_header = 18
193 fs_axis = 12

```

```
194     fs_ticks = 10
195     fs_legend = 10
196     Opacity_Image = 0.7
197
198     fig, ax = plt.subplots(figsize=(6, 6))
199     ax.set_xlim(-125, 125)
200     ax.set_ylim(-125, 125)
201
202
203     img = colour.read_image('F://!Aufgaben/001_aktuelle Aufgaben/
Masterarbeit/Python/Assets/Messung/M11_Spektraleigenschaften/
LAB_Background.jpg')
204
205     ax.imshow(img, extent=[-125.7, 126.7, -126.6, 125.6], aspect='
auto', alpha=Opacity_Image)
206
207     ax.set_xlabel("b*", fontsize=fs_axis)
208     ax.set_ylabel("a*", fontsize=fs_axis)
209
210     ax.set_xticks
([ -100, -75, -50, -25, 0, 25, 50, 75, 100 ], [ -100, -75, -50, -25, 0, 25, 50, 75, 100 ],
fontsize=fs_ticks)
211     ax.set_yticks
([ -100, -75, -50, -25, 0, 25, 50, 75, 100 ], [ -100, -75, -50, -25, 0, 25, 50, 75, 100 ],
fontsize=fs_ticks)
212
213     ax.scatter(measured_display_linear_Lab[:, 2],
measured_display_linear_Lab[:, 1], color="black", label="ITU-R BT
.2100 HLG Reference Camera")
214
215     ax.scatter(uc4000_Lab[:, 2], uc4000_Lab[:, 1], color="red", label="
Panasonic UC4000")
216
217     for i in range(len(measured_display_linear_Lab[:, 1])):
218         ax.quiver(uc4000_Lab[i, 2], uc4000_Lab[i, 1],
measured_display_linear_Lab[i, 2] - uc4000_Lab[i, 2],
measured_display_linear_Lab[i, 1] - uc4000_Lab[i, 1], angles='xy',
scale_units='xy', scale=1, color='#FFFFFF', width=0.005)
219
220
221     ax.legend(fontsize=fs_legend, loc='lower center')
222
223
224     plt.show()
225
226
227     fig, (ax1, ax0, ax2, ax3) = plt.subplots(4, figsize=(12, 6),
gridspec_kw = {'height_ratios': [4, 0.4, 1, 1]})
228
229
230     fs_header = 18
231     fs_axis = 12
232     fs_ticks = 10
233     fs_legend = 10
234     Opacity_Image = 0.5
235
```

```

236     ax1.set_title('Relative Display-Luminanz',fontsize = fs_header ,
237                 pad = 20)
238     ax2.axis("off")
239     ax3.axis("off")
240     ax0.axis("off")
241     position = np.linspace(0, len(measured_display_linear_Lab[:,0])
242                            -1, len(measured_display_linear_Lab[:,0]))
243
244     xyz_reference = np.array(xyz_reference)
245     for i in range(24):
246         color=np.clip(colour.models.XYZ_to_RGB(xyz_reference[i,:]/
247 np.max(xyz_reference),"ITU-R BT.709"),0,1)
248         color=colour.models.eotf_inverse_BT1886(color)
249         ax2.bar(position[i], 1, width=0.8, color=color, alpha=1)
250
251     data = np.random.rand(24)
252     labels = [f'Nr. {i+1}' for i in range(24)]
253
254     plt.figure(figsize=(10,6))
255     ax1.set_ylim(0, 1)
256     ax1.bar(position, measured_display_linear_Lab[:,0]/100, width
257            =0.8, alpha=0.5, color='black', label="ITU-R BT.2100 HLG Reference
258            Camera")
259     ax1.bar(position, uhkx700_Lab[:,0]/100, width=0.8, color='red',
260            alpha=0.8, label="Panasonic UHKX70")
261
262     ax1.set_xticks(ticks=range(len(data)), labels=labels, rotation
263            =90, ha='center',fontsize=fs_ticks)
264     ax1.set_yticks([0,0.25,0.50,0.75,1],[0,25,50,75,100],fontsize=
265            fs_ticks)
266
267     ax0.text(0.5, -4.6, 'Farbtafel Nummer',fontsize=fs_axis,
268            horizontalalignment='center')
269     ax1.set_ylabel('Relative Luminanz [%]',fontsize=fs_axis)
270
271     ax1.legend(fontsize=fs_legend,loc='lower center',
272            bbox_to_anchor=(0.5, -1, 0, 0))
273
274     plt.tight_layout()
275     plt.show()

```

Listing D.12: Farbtreue der Kameras