



Institut für Visualisierung und Interaktive Systeme

Universität Stuttgart
Universitätsstraße 38
D-70569 Stuttgart

Diplomarbeit Nr. 3596

HDR Beleuchtung von realen Objekten mit einem LDR Bildschirm für fotografische Anwendungen

Manuel Jerger

Studiengang: Informatik

Prüfer/in: Jun.-Prof. Martin Fuchs

Betreuer/in: Sebastian Koch, M.Sc.

Beginn am: 16. Dezember 2013

Beendet am: 17. Juni 2014

CR-Nummer: I.3.3, I.4.1

Kurzfassung

In dieser Arbeit wird untersucht, wie man mit einem handelsüblichen Flachbildschirm eine Szene fotografisch beleuchten kann. Spiegelnde und transparente Objekte lassen sich in einem Fotostudio nur eingeschränkt künstlich beleuchten, da sich die gesamte Umgebung in ihnen spiegelt und bricht. Sie werden deshalb üblicherweise in einem weißen Fotozelt mit diffusen Flächenlichtquellen beleuchtet. Verwendet man stattdessen einen Flachbildschirm als Lichtquelle, so können diese Objekte auch unter einer detaillierten Umgebungsbeleuchtung fotografiert werden. Das Kontrastverhältnis herkömmlicher Bildschirmtechnologien ist im Vergleich zu dem realer Beleuchtungsszenarien allerdings sehr klein. Zur Abhilfe kann man statt einem statischen Bild eine zeitvariante Sequenz einsetzen, womit sich die kleinste darstellbare Lichtmenge reduzieren, und der Dynamikbereich vergrößern lässt. In dieser Arbeit wird ein System vorgestellt, das es ermöglicht, eine Szene mit einem Bildschirm nahezu vollständig, in hoher Auflösung und mit hohem Dynamikbereich zu beleuchten. Als Lichtquelle kommt dabei ein Laptop mit einer eingebauten Webcam zum Einsatz, sodass eine Beleuchtung von Hand erzeugt werden kann. Ein Benutzer bewegt ihn dazu auf einer Halbkugel rund um eine Szene und erzeugt stückweise einzelne Teile der einfallenden Beleuchtung. Die Bildschirmposition wird dabei automatisch mit der Webcam anhand von geometrischen Markern berechnet, die rund um die Szene angeordnet sind. Die einzelnen Teilbeleuchtungen werden von einer Digitalkamera fotografiert, und die Aufnahmen im Anschluss zu einer vollständig beleuchteten Szene aufaddiert. Es wird dabei unter Anderem auch gezeigt wie sich der Dynamikbereich der Beleuchtung durch Einsatz einer zeitvarianten Sequenz erhöhen lässt, und wie sich solche Sequenzen konstruieren lassen.

Inhaltsverzeichnis

1. Einleitung	7
1.1. Aufgabenstellung	8
1.2. Gliederung	9
2. Grundlagen	11
2.1. Notationen	11
2.2. Licht und Beleuchtung	12
2.2.1. Enviroment-Mapping	12
2.2.2. Light-Stages	13
2.3. Dynamikbereich	14
2.4. Bildschirme	16
2.5. Kameras	18
2.5.1. Positions berechnung mit Kameras	19
3. Verwandte Arbeiten	21
3.1. Light-Stages	21
3.2. Environment-Matting	22
4. Modell und Kalibrierung	25
4.1. Das geometrische Modell	26
4.1.1. Kamera-Parameter	27
4.2. Das radiometrische Modell	28
4.2.1. Ansprechverhalten der Kamera	30
4.2.2. Ansprechverhalten des Bildschirms	31
5. Tracking-Stage	37
5.1. Auswahl der Fiducials	38
5.2. Konstruktion	39
5.3. Evaluation: Tracking-Stage	41
6. Beleuchtung	43
6.1. Benötigte Strahldichte	44
6.2. LDR-Beleuchtung	45
6.2.1. Reduktion von L_0	47
6.3. HDR-Beleuchtung	48
6.4. Vollständige Beleuchtung	51
6.4.1. Behandeln von Überlappungen	52

6.4.2. Ablauf	53
6.4.3. Zusammenfügen der Aufnahmen	56
6.5. Evaluation	56
6.5.1. Evaluation: LDR-Beleuchtung	57
6.5.2. Evaluation: HDR-Beleuchtung	58
6.5.3. Evaluation: Überlappungen	61
7. Ergebnisse	63
7.1. Ergebnis: Szene 1	63
7.2. Ergebnis: Szene 2	68
8. Zusammenfassung und Ausblick	71
A. Anhang	73
A.1. Hardware	73
A.1.1. Laptop	73
A.1.2. DSLR-Kamera	73
A.1.3. Tracking-Stage Komponenten	74
A.2. Software	74
A.3. Ergänzende Abbildungen	75
A.4. Symbolverzeichnis	76
Literaturverzeichnis	77

Abbildungsverzeichnis

1.1.	Eine reale Szene mit einen Bildschirm beleuchten	8
2.1.	Beleuchtung durch weit entfernte Punktlichtquellen	13
2.2.	Aufbau einer Cube-Map	14
2.3.	Eine Light-Stage	15
2.4.	Blickwinkelabhängigkeit bei einem TN-Bildschirm	17
2.5.	Abgenutzte Pixel eines LED-Bildschirms	17
2.6.	Funktionsweise einer idealen Kamera	18
2.7.	Positionsberechnung mit einer Kamera	19
4.1.	Anordnung der Komponenten im System	26
4.2.	Geometrisches Modell	27
4.3.	Lichtfluss zwischen Bildschirm, Szene und Kamera	29
4.4.	Blockdiagramm der radiometrischen Beziehungen	30
4.5.	Ansprechverhalten einer DSLR-Kamera	31
4.6.	Aufbau bei der radiometrische Kalibrierung des Bildschirms	32
4.7.	Aufteilung der Bildschirmfläche	34
4.8.	Lichtverteilung des Bildschirms	34
4.9.	Ansprechverhalten des Bildschirms	35
5.1.	Fiducial-Ebenen auf der Tracking-Stage	37
5.2.	Generierte Fiducials	38
5.3.	Fiducials der Testreihe	39
5.4.	Evaluation der Fiducialgröße	40
5.5.	Aufbau der Tracking-Stage	41
5.6.	Fotos der Tracking-Stage	41
5.7.	Evaluation der Tracking-Stage	42
6.1.	Projektion von Cube-Map auf Bildschirmebene	44
6.2.	Projektion von Cube-Map-Seite auf Bildschirmebene	45
6.3.	Interpolation des Per-Pixel-Ansprechverhaltens	46
6.4.	1D-Beispiel einer HDR-Sequenz	49
6.5.	HDR-Frames: Uniforme Strahldichte	51
6.6.	HDR-Frames: Echte Szene	51
6.7.	Flussdiagramm: Überlappungsbehandlung	53
6.8.	Flussdiagramm: Beleuchtungsablauf	54
6.9.	Haltung des Benutzers bei der Beleuchtung	55

6.10. Evaluation der LDR-Beleuchtung (Ursprung): Heatmap	57
6.11. Evaluation der LDR Beleuchtung (Winkel): Aufnahmewinkel	58
6.12. Evaluation der LDR-Beleuchtung (Winkel): Diagramme	59
6.13. Evaluation der HDR-Sequenz	60
6.14. Evaluation der Überlappung von Teilbeleuchtungen	61
7.1. Ergebnis: Szene 1 (Cube-Map)	64
7.2. Ergebnis: Szene 1 (Bildschirmpositionen als 3D-Plot)	64
7.3. Ergebnis: Szene 1	65
7.4. Ergebnis: Szene 1 (Artefakte)	66
7.5. Ergebnis: Szene 1 (vier Teilbeleuchtungen)	67
7.6. Ergebnis: Szene 1 (Vergleich)	67
7.7. Ergebnis: Szene 1 (Hintergrund)	68
7.8. Ergebnis: Szene 2	69
A.1. Ansprechverhalten der DSLR im Log-Log Plot	75
A.2. Fiducials der Tracking-Stage	75
A.3. Formel zur Luminanzberechnung	76

Tabellenverzeichnis

2.1. Matrix- und Vektoroperationen	11
7.1. Ergebnis: Szene 1 (Laufzeiten)	65
A.1. Technische Daten des Laptops	73
A.2. Technische Daten der DSLR-Kamera	73
A.3. Komponenten der Tracking-Stage	74
A.4. Verwendete Softwarepakete und Bibliotheken	74

1. Einleitung

Die Herstellung von Produktabbildungen für Werbung und Kataloge erfolgt oft in einem Fotostudio unter Einsatz einer sehr aufwändigen Beleuchtungstechnik. Das Erscheinungsbild eines Gegenstandes wird nämlich nicht nur durch die Form und das Material bestimmt, sondern hängt auch davon ab wie es beleuchtet wird. Dies ist insbesondere bei spiegelnden oder transparenten Materialien der Fall: Ihr Aussehen wird weitgehend durch die einfallende Beleuchtung bestimmt, da sich an ihnen die gesamte Umgebung reflektiert oder bricht. Das Licht, das von einer realen Umgebung ausgeht, besitzt sehr feine Details und einen hohen Kontrast, weshalb es in einem Fotostudio nicht ohne Weiteres erzeugt werden kann. Objekte dieser Art werden darum in der Regel in einem weißen Fotozelt fotografiert, das eine gleichmäßige, diffuse Beleuchtung erzeugt.

Soll das Objekt jedoch unter einer realen Beleuchtung abgebildet werden, so muss man es entweder direkt unter einer realen Umgebungsbeleuchtung fotografiert, oder mit Hilfe eines Computermodells rendern. Das Rendern ist oft die einfachste und kostengünstigste Möglichkeit, es erfordert jedoch dass eine Beschreibung der Form und der Materialeigenschaften vorliegt, was nicht immer der Fall ist. Bei komplizierten, natürlichen Objekten ist dies auch teilweise gar nicht möglich, da die tatsächliche Form aber auch das Verhalten des Lichts im Material nicht bekannt ist, und auch nicht ohne großen Aufwand gemessen werden kann. Aus diesem Grund wird ein Verfahren benötigt, mit dem man solche Materialien in einem Fotostudio kostengünstig und mit hoher Auflösung beleuchten kann.

Eine hochauflösende Beleuchtung erfordert sehr viele, individuell steuerbare Lichtquellen, die rund um das Objekt angeordnet sind. Hierzu sind Flachbildschirme ideal geeignet, denn sie bestehen aus einem Array von hundert tausenden einzelnen, farbigen Lichtpunkten und lassen sich präzise und mit hoher Frequenz ansteuern. Mit einem einzelnen Bildschirm lässt sich aber nur ein Teil einer umschließenden Beleuchtung erzeugen. Bewegt man ihn an viele Positionen rund um das Objekt, so kann man eine vollständige Beleuchtung stückweise erzeugen. Fotografiert man die Szene unter vielen Teilbeleuchtungen, so kann man die Aufnahmen im Anschluss kombinieren und erhält eine vollständig beleuchtete Szene.

Für eine realistische Beleuchtung wird Licht mit einem hohen Kontrastverhältnis benötigt. Gewöhnliche Bildschirme, wie sie in den meisten portablen Geräten verbaut sind, besitzen jedoch nur einen geringen Dynamikbereich, der weitaus kleiner als der einer realen Beleuchtung ist. Sie können also nicht ohne Weiteres direkt für eine fotografische Beleuchtung verwendet werden. Mit Wissen über die Funktionsweise eines Bildschirms ist es jedoch möglich, seinen Dynamikbereich durch eine zeitvariante Bildsequenz zu erhöhen.

1. Einleitung

1.1. Aufgabenstellung

In dieser Arbeit soll untersucht werden, wie sich ein Bildschirm mit handelsüblichen Kontrast zum Erzeugen einer fotografischen Beleuchtung einsetzen lässt. Die Beleuchtung soll dabei neben einer hohen Auflösung auch einen großen Dynamikbereich besitzen. Das Ziel ist es, dass der Bildschirm von Hand rund um die Szene bewegt werden kann. Seine Koordinaten sollen dabei automatisch durch eine daran befestigte Kamera bestimmt werden. An jeder Position soll, wie in Abbildung 1.1 dargestellt, ein Ausschnitt (B_1) einer vorgegebenen Beleuchtung erzeugt und fotografiert werden. Die einzelnen Teilbeleuchtungen müssen dabei so erzeugt werden, dass sich die einzelnen Aufnahmen im Anschluss zu einer korrekten, vollständigen Beleuchtung zusammenführen lassen. Es soll also eine zusammenhängende Lichtfläche entstehen, weshalb die Überlappung zu aufeinander folgende Teilbeleuchtungen (B_2, B_3) beachtet werden muss: Der Bildschirm darf immer nur das Licht emittieren, das noch nicht von einer vorangegangenen Teilbeleuchtung hergestellt wurde.

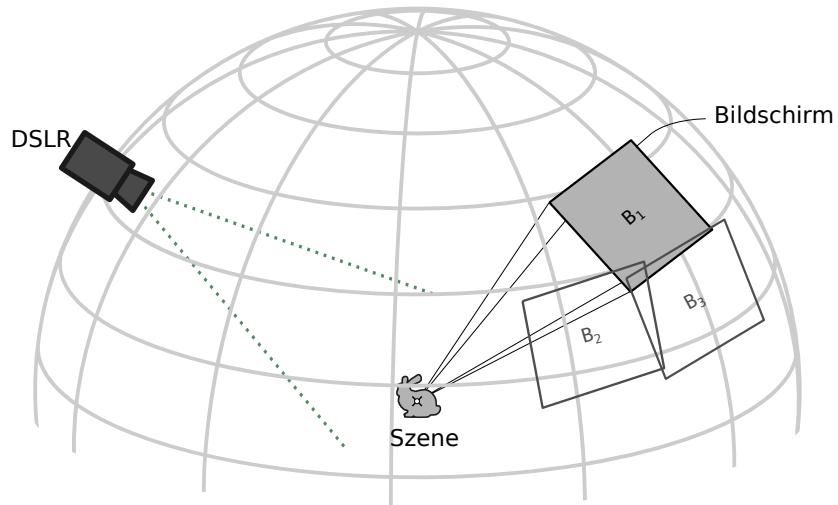


Abbildung 1.1.: Ein Objekt wird mit einem Bildschirm beleuchtet, der an mehrere Stellen $B_1 – B_3$ bewegt wird und dort jeweils einen Teil der einfallenden Beleuchtung erzeugt.

Der Bildschirm und die Kameras müssen dazu sowohl radiometrisch als auch geometrisch kalibriert werden, sodass mit den Bildschirmpixeln gezielt eine einfallende StrahldichteVerteilung erzeugt werden kann. Die Beleuchtungsberechnung soll automatisiert erfolgen, sodass der Benutzer den Bildschirm lediglich an mehrere Positionen, wie auf einer Halbkugel, rund um das Objekt bewegen muss. Da der Dynamikbereich eines herkömmlichen Bildschirms klein ist, soll des weiteren untersucht werden, ob und wie man eine zeitvariante Sequenz dazu verwenden kann, ihn zu vergrößern, sodass eine natürliche Umgebungsbeleuchtung hergestellt werden kann. Dabei sollen keine teuren Messgeräte oder aufwändige Aufbauten eingesetzt werden.

1.2. Gliederung

Die Arbeit ist in folgender Weise gegliedert:

2. Grundlagen

Erklärt die wichtigsten Grundlagen, die zum Verständnis der Arbeit benötigt werden: Die verwendeten mathematischen Notationen; Licht und distante Beleuchtung, Cube-Maps, Dynamikbereich von Licht; das Pinhole-Modell für Kameras sowie seine Anwendung bei der Positionsbestimmung; aktuelle Bildschirmtechnologien und ihre Eigenschaften mit Hinblick auf die Beleuchtungserzeugung.

3. Verwandte Arbeiten

Verwandte Arbeiten die sich mit der Beleuchtung von Szenen auseinandersetzen.

4. Modell und Kalibrierung

Hier wird das System und alle Komponenten zuerst vorgestellt, dann radiometrisch sowie geometrisch modelliert und anschließend kalibriert.

5. Tracking-Stage

Damit die Position des Bildschirms mit einer Kamera bestimmt werden kann, müssen geometrische Marker um die Szene angeordnet werden. Dieser Aufbau wird als "Tracking-Stage" bezeichnet. In diesem Kapitel wird erklärt wie die Markierungen ausgewählt und aufgebaut wurden. Zum Abschluss wird die Genauigkeit der Positionsrechnung evaluiert.

6. Beleuchtung

Der Kernpunkt der Arbeit: Es wird gezeigt, wie mit einem radiometrisch kalibrierten Bildschirm ein Teil einer vorgegebenen Beleuchtung erzeugt werden kann. Daraufhin wird die Theorie hinter der zeitvarianten Sequenz erklärt, und gezeigt, dass es damit möglich ist das Kontrastverhältnis zu erhöhen. Im Anschluss wird ein Verfahren vorgestellt, das es einen Benutzer ermöglicht eine zusammenhängende, nahezu vollständige Beleuchtung mit einem handgehaltenen Bildschirm zu erzeugen. Eine Evaluation der einzelnen Aspekte des Verfahrens bildet den Abschluss des Kapitels.

7. Ergebnisse

Erzeugen einer vollständigen und einer HDR-Beleuchtung mit dem vorgestellten Verfahren, Diskussion der Ergebnisse.

8. Abschluss

Die Zusammenfassung der Arbeit und ein Ausblick auf mögliche Erweiterungen des Verfahrens.

2. Grundlagen

In diesem Kapitel werden die wichtigsten Grundlagen erklärt, die zum Verständnis dieser Arbeit notwendig sind. Als erstes werden die *mathematischen Notationen* der verwendeten Symbole und Operationen, insbesondere Matrixoperatoren, eingeführt. Daraufhin wird auf das Thema *Beleuchtung* eingegangen und erklärt wie sich die Beleuchtung, die auf eine Szene einfällt, mathematisch darstellen lässt. Es wird das Konzept des *Dynamikbereichs* erklärt, mit dem das Kontrastverhältnis einer Beleuchtungen sowie von Lichterzeugenden und -aufnehmenden Geräten beschrieben werden kann. Des Weiteren wird das *Pinhole-Modell* vorgestellt, mit dem Kameras geometrisch modelliert werden können, sodass mit ihnen eine *Positionsbestimmung* möglich ist. Zuletzt werden die Bildschirmtechnologien betrachtet, die in gängigen portablen Geräten zu finden sind, und gezeigt worin die Unterschiede im Hinblick auf die Lichterzeugung liegen.

2.1. Notationen

Diese Arbeit hält sich an die üblichen mathematischen Konventionen: Skalare Werte und Funktionen werden mit Kleinbuchstaben bezeichnet, für Vektoren und vektorwertige Funktionen wird ein Pfeil übergestellt. Matrizen, wozu auch Rasterbilder gehören, werden mit Großbuchstaben bezeichnet. Ein Element einer Matrix, also auch ein Pixel eines Bildes, wird mit ganzzahligen Koordinaten (x, y) adressiert, wobei $(0, 0)$ das Element in der linken oberen Ecke bezeichnet.

In Tabelle 2.1 sind alle verwendeten Matrix- und Vektoroperationen aufgeführt. Damit soll klargestellt werden, bei welchen Operationen es sich um Elementweise Operationen handelt, da sie bei der Behandlung von Bildern häufig benötigt werden.

aB	Skalarmultiplikation
AB	Matrixmultiplikation
$A\vec{b}$	Matrix-Vektor Multiplikation
$A \cdot B$	Elementweise Multiplikation
A/B	Elementweise Division
$A(i, j)$	i -tes Element von links in der j -ten Zeile von oben
$A \leq B$	Elementweiser Vergleich: muss für alle Elemente gelten
$A = n$	Alle Elemente setzen oder vergleichen
$\max(A), \min(A)$	Das größte / kleinste Element

Tabelle 2.1.: Notationen der verwendeten Matrixoperationen. Alle elementweise arbeitende Operationen und Funktionen gelten auch für Vektoren.

2.2. Licht und Beleuchtung

In jeder realen Umgebung befinden sich Lichtquellen, die dafür sorgen, dass sie durch und durch von Licht durchdrungen ist. Eine kleine Szene die sich darin befindet, wie beispielsweise ein einzelnes Objekt, wird durch das Licht, das von der umgebenden Materie ausgeht, rundum beleuchtet. Wenn die Szene selbst keine Lichtquellen beinhaltet so kann man die Umgebung als die einzige Quelle betrachten. Unter der Annahme, dass der Ursprung des Lichts das die Szene erreicht, unendlich weit von ihr entfernt ist, ist eine mathematische Beschreibung der einfallenden Beleuchtung einfach möglich: Das Licht, das in einem Punkt der Szene eingeht, kann mit einer Funktion

$$l_{dist}(\phi, \theta) \quad (2.1)$$

beschrieben werden, welche die aus der Richtung (ϕ, θ) eingehende Strahldichte beschreibt.

Dieser Sachverhalt wird in dieser Arbeit als “distanter Beleuchtung” (engl: “distant illumination”) bezeichnet. Eine distante Beleuchtung kann man mit unendlich weit entfernten Punktlichtquellen modellieren, die kugelförmig rund um die Szene angeordnet sind. In der Praxis ist eine distante Beleuchtung natürlich nicht realisierbar. Ist die Entfernung der Lichtquellen jedoch deutlich größer als der Radius der Szene (Abbildung 2.1), so kann eine distante Beleuchtung approximiert werden. Beispielsweise weicht bei einem Verhältnis von $r_l/r_s > 10$ die Richtung der einfallenden Lichtstrahlen um nicht mehr als $\text{atan}(0.1) \approx 5.71^\circ$ ab. Vergrößert man den Abstand oder reduziert den Szeneradius, so verbessert sich die Approximation entsprechend.

Hier sei anzumerken dass mit dem Begriff “Strahldichte” in dieser Arbeit ausschließlich *relative* Strahldichte gemeint ist. Sie ist proportional zur absoluten Strahldichte, hängt jedoch mit ihr über einen unbekannten Faktor zusammen. Aufgrund der linearen Eigenschaft von Lichttransport darf die relative Strahldichte in allen linearen Rechenoperationen verwendet werden, solange kein direkter Bezug zu absoluten Messwerten bestehen muss.

Das Prinzip der distanten Beleuchtung hat zahlreiche Anwendungen in der Computergrafik und anderen dazu verwandten Gebieten. Im Folgenden werden zwei davon vorgestellt, die im Zusammenhang zu dieser Arbeit stehen.

2.2.1. Environment-Mapping

Das Environment-Mapping-Verfahren und die dabei verwendeten Environment-Maps wurden im Jahr 1976 von Blinn und Newell vorgestellt [BN76]. Ihr Ziel dabei war das Rendern spiegelnder Objekte mit vorwiegend konvexen Oberflächen, was mit den Methoden dieser Zeit noch nicht möglich war. Spiegelnde Oberflächen reflektieren die gesamte Umgebung, inklusive allen anderen Objekte die sich darin befinden, weshalb Strahlverfolgenden Verfahren wie zum Beispiel Raytracing eingesetzt werden müssen. Mit dem Environment-Mapping haben sie eine Methode vorgestellt, die ohne großen Aufwand verblüffend realistische Ergebnisse erzielt. Sie nehmen dabei eine distante Beleuchtung an und speichern das in die Szene einfallende Licht in einer Environment-Map. Dabei handelt es sich um eine gewöhnliche zweidimensionale Textur, wobei eine der Dimensionen für die Darstellung des Polarwinkels ϕ und eine für den Azimutalwinkel θ der Einfallsrichtung verwendet wird. Die

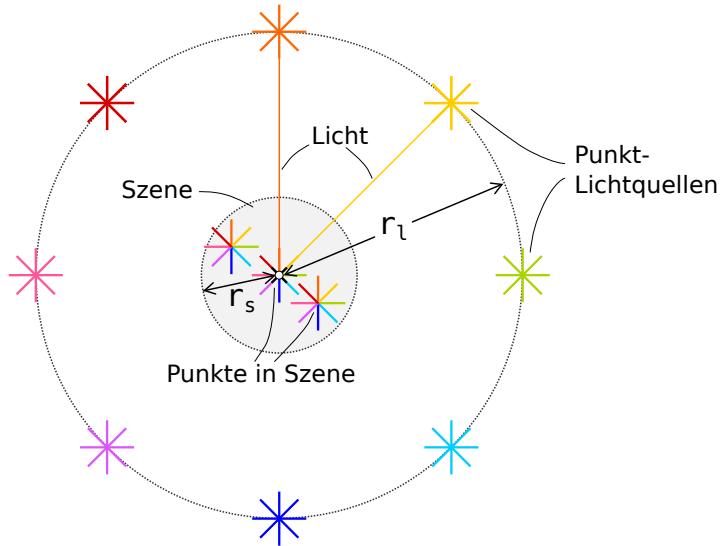


Abbildung 2.1.: Eine Szene mit dem Radius r_s wird von mehreren Punktlichtquellen beleuchtet, die sich in einem Abstand von r_l zum Szenenmittelpunkt befinden. Wenn der Quotient r_l/r_s groß genug ist, so kann davon ausgegangen werden, dass in allen Punkten der Szene aus jeder Richtung das selbe Licht einfällt. Damit lässt sich eine Approximation der einfallenden Beleuchtung mit Gleichung 2.1 darstellen.

Intensität des Lichts, das von jedem Punkt des Objekts zur Kamera reflektiert wird, kann dann ganz einfach berechnet werden: Anhand der Oberflächennormale an einem Punkt lässt sich der Reflexionsvektor berechnen, womit die Richtung des einfallenden Lichstrahls bekannt ist. Seine Strahldichte kann dann direkt durch Nachschlagen des zugehörigen Pixels in der Environment-Map ermittelt werden. Damit lassen sich mit sehr geringem Aufwand realistische Ergebnisse erzielen, weshalb Environment-Mapping auch heute noch häufig in der Computergrafik eingesetzt wird.

Die von Blinn und Newell verwendete winkelparametrisierte Environment-Map besitzt einige Nachteile, wie beispielsweise eine stark ungleichmäßige Auflösung und rechenaufwändige Winkeloperationen, weshalb im Laufe der Zeit praxistauglichere Parametrisierungen entwickelt wurden. Eine häufig anzutreffende Darstellungsform ist die *Cube-Map* [AMHH11]. Man kann sie sich als ein Würfel aus sechs Texturen vorstellen, der rund um eine Szene aufgespannt ist (Abbildung 2.2). Die Pixel der Texturen speichern dabei die Strahldichte eines Lichstrahls, und ihre Koordinaten auf der Textur geben die Lichtrichtung an.

2.2.2. Light-Stages

Eine *Light-Stage* kann man als eine reale Implementierung einer Environment-Map ansehen. In den meisten Fällen besteht sie aus zahlreichen, einzeln steuerbaren Lampen, die kugelförmig rund um eine Szene angeordnet sind (Abbildung 2.3). Wenn Szenendurchmesser und Lichtquellenabstand so gewählt werden, dass die Annahme einer distanten Beleuchtung erfüllt wird, dann kann man die Lampen als Punktlichtquellen ansehen, ihre Helligkeit durch Abtasten einer Environment-Map

2. Grundlagen

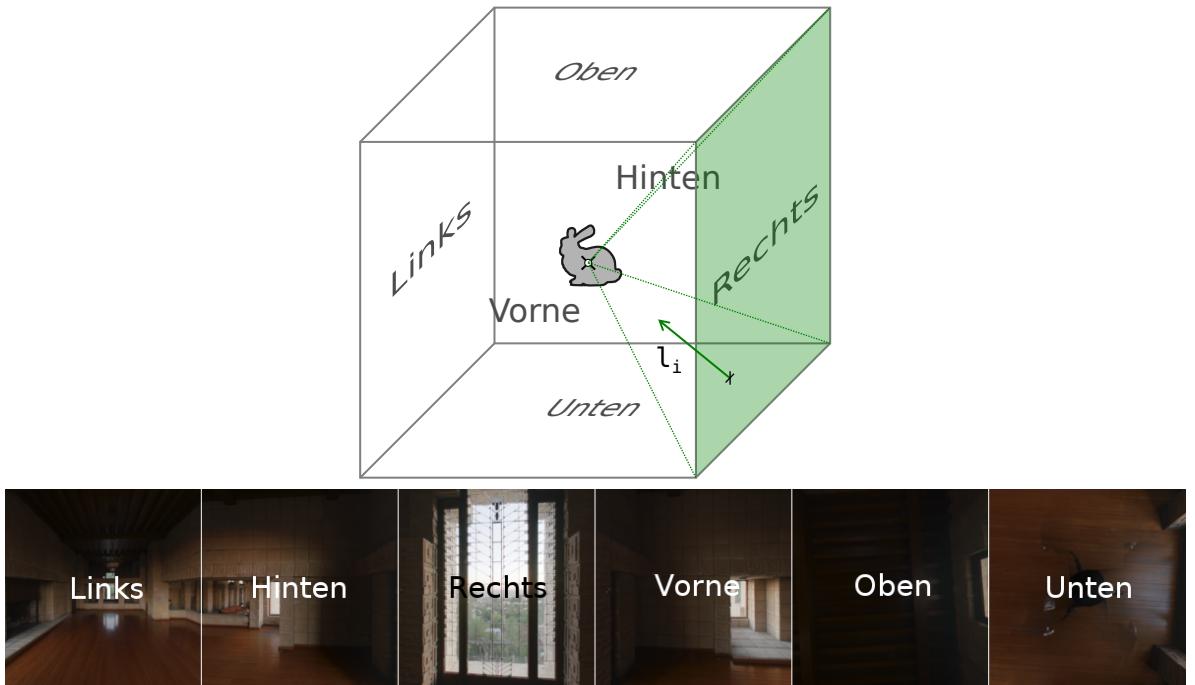


Abbildung 2.2.:

Oben: Eine Cube-Map besteht aus sechs Texturen, die man sich rund um eine Szene angeordnet vorstellen kann. Ein Pixel speichert dabei die in den Ursprung einfallende Strahldichte l_i , und seine Koordinaten bestimmen die Richtung. Unten: Die sechs Texturen einer Cube-Map, nebeneinander dargestellt. Sie wurden anhand der "Ennis-Brown House"-Environment-Map aus der Debevec High-Res Light Probe Gallery [04] berechnet.

berechnen und eine einfallende Beleuchtung erzeugen. Die Auflösung dieser Beleuchtung ist jedoch stark eingeschränkt, da die Lampen nicht beliebig weit entfernt positioniert oder beliebig klein gebaut werden können. Aus diesem Grund eignen sich Light-Stages nicht zur Beleuchtung spiegelnder Objekte, denn man würde die einzelnen Lampen auf der Oberfläche erkennen können. Für diffuse Szenen sind sie hingegen sehr gut geeignet, weshalb sie in der Forschung zahlreiche Anwendungen haben (siehe Abschnitt 3.1)

2.3. Dynamikbereich

Der *Dynamikbereich* (auch: Kontrastverhältnis) beschreibt das Verhältnis zwischen dem größten und dem kleinsten Wert größer 0, welches ein Gerät oder System speichern, darstellen, aufnehmen oder allgemein handhaben kann. Er kann, je nach Anwendungsbereich, auf jede beliebige messbare

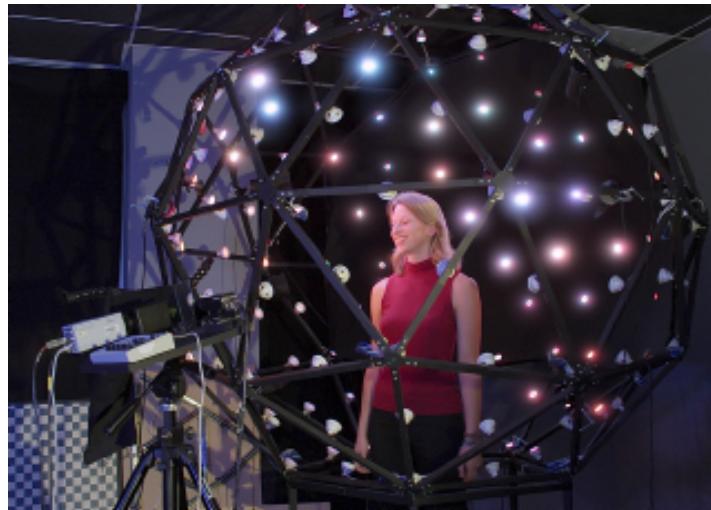


Abbildung 2.3.: Die “Light-Stage 3” von Debevec et al. erzeugt eine einfallende Beleuchtung mit Hilfe zahlreicher computergesteuerter Lampen (Bildquelle: [DWT⁺02])

technische Größen (Licht, Schalldruck, Strahlung usw.) bezogen werden. Für diese Arbeit wird der Dynamikbereich R über die relative Strahldichte definiert und lautet

$$R = \frac{l_{max}}{l_{min}} \quad \text{wobei } l_{min} > 0 \quad (2.2)$$

wobei l_{max} und l_{min} die kleinste beziehungsweise die größte Strahldichte bezeichnet. Damit lässt sich der Dynamikbereich einer Umgebungsbeleuchtung oder einer Aufnahme berechnen.

Soll jedoch der Dynamikbereich eines Bildschirms oder einer Kamera ermittelt werden, so muss der kleinste beziehungsweise größte Wert verwendet werden, der *mit jedem* Pixel darstellbar ist. Ein Beispiel: Ein Bildschirm, bei dem jedes Pixel in der Lage ist $l_{max} = 100.0$ und $l_{min} = 1.0$ zu erzeugen, hat einen Dynamikbereich von $R = 100$, auch geschrieben als $100 : 1$

Es sei anzumerken dass das Kontrastverhältnis eines Flachbildschirms vom Hersteller üblicherweise über die Leuchtdichte, und nicht die Strahldichte angegeben wird. Da der Bildschirm in dieser Arbeit jedoch nicht von einem Menschen betrachtet wird, sondern zum Erzeugen einer StrahldichteVerteilung in einem Punkt verwendet wird, ist die Definition in Gleichung 2.2 zu verwenden. R kann darum nicht mit den Herstellerangaben verglichen werden.

Der Dynamikbereich spielt eine wichtige Rolle in der Fotografie: Die Strahldichte in einer realen Szene kann um mehr als 10 Größenordnungen variieren [RWP05] - deutlich mehr als man mit einer Kamera in einem Bild aufnehmen kann. Man spricht hier jeweils von High Dynamic Range ("HDR") und Low Dynamic Range ("LDR"), wobei Digitalkameras und Bildschirme zu LDR gezählt werden, und eine reale, Sonnenbeleuchtete Szene dagegen zu HDR [RWP05].

Das Menschliche Auge kommt sehr gut mit dem großen Dynamikbereich einer natürlichen Szene zurecht, und kann bis zu 5 Größenordnungen gleichzeitig wahrnehmen [RWP05]. Damit auf einem Bildschirm eine echte Szene realistisch dargestellt werden kann, muss er also auch einen

2. Grundlagen

entsprechend hohes Kontrastverhältnis besitzen. In der Praxis ist dies nicht der Fall: Nahezu alle Bildschirme sind LDR, da für ein HDR-Display ein deutlich höherer technischer Aufwand erforderlich ist. HDR-Bildschirme und HDR-Kameras sind aktuell noch Forschungsgegenstand und nicht im Konsumerbereich anzutreffen.

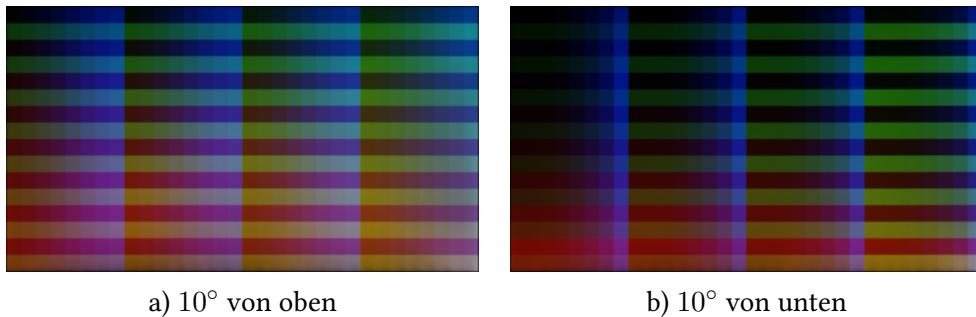
Im Folgenden werden die aktuell verbreiteten Bildschirmtechnologien vorgestellt, und die Unterschiede im Hinblick auf die Erzeugung einer (HDR-) Beleuchtung dargelegt.

2.4. Bildschirme

Die Entwicklung von Flachbildschirmen ist immer noch in vollem Gange und es ist noch lange kein Ende in Sicht. Die wichtigste Technologie stellt dabei seit langem der Flüssigkristall dar, der die Basis für alle LCD-Bildschirme (LCD, TFT) bildet. Ein Flüssigkristallbildschirm nutzt den Effekt der Polarisierung, womit die Lichtmenge geregelt werden kann die ein einzelnes Pixel transmittiert. Das Licht, das von diesen Bildschirmen ausgeht, wird von einer gemeinsamen Hintergrundbeleuchtung erzeugt und durch die mit einem Farbfilter versehenen Pixel moduliert. Diese Hintergrundbeleuchtung befindet sich in der Regel im Rahmen des Bildschirms und wird durch einen Lichtleiter über die gesamte Bildschirmfläche verteilt. Sie ist, besonders bei einer kompakten Bauweise in mobilen Geräten, sehr unregelmäßig und variiert über die Oberfläche.

Der Dynamikbereich eines Bildschirms wird zum Einen von der maximalen Helligkeit der Hintergrundbeleuchtung, und zum Anderen von dem Regelbereich der Pixel limitiert. LCD-Bildschirme haben die intrinsische Eigenschaft, dass ein Pixel nicht vollständig auf "undurchlässig" geschaltet werden kann - es geht zwangsläufig immer Licht von den Pixeln aus. Das sorgt für den hohen Schwarzwert der für diese Bildschirme typisch ist.

Des Weiteren sind LCD-Bildschirme stark Blickwinkelabhängig - die maximale Strahldichte, aber auch das Ansprechverhalten des Bildschirms, hängt maßgeblich von der Richtung ab, unter der das Licht die Oberfläche verlässt. Bedingt durch die Konstruktion der Pixel ist dieser Effekt besonders bei älteren und kostengünstigen Bildschirmen ausgeprägt (TN-Technologie), konnte jedoch durch andere Bauweisen verbessert werden (IPS-Technologie). Abbildung 2.4 zeigt diesen Effekt anhand eines TN-Bildschirms, der aus zwei unterschiedlichen Blickwinkel fotografiert wurde.


 a) 10° von oben

 b) 10° von unten

Abbildung 2.4.: Ein $3 \times 3 \times 3$ Bit Unterraum des möglichen $8 \times 8 \times 8$ -Bit Farbraums wurde auf einem TN-Bildschirm dargestellt, und unter zwei verschiedenen Blickwinkeln aus 2.5 Meter Entfernung fotografiert. Es ist gut zu sehen dass sich das Ansprechverhalten des Bildschirms sich dabei verändert.

Eine relativ neue Technologie, die diese Probleme nicht besitzt, ist der LED-Bildschirm (“AMOLED”, “OLED”), der auf einem ganz anderen Ansatz basiert: Die Pixel werden dabei durch einzelne Leuchtdioden (“LED”) bestehen also aus diskreten Lichtquellen. Damit ist es möglich, einen Pixel komplett auszuschalten, weshalb diese Bildschirme einen hervorragenden Schwarzwert und somit eine größeres Kontrastverhältnis als LCD-Bildschirme besitzen. Zudem ist das abgestrahlte Licht weitaus weniger Richtungsabhängig - ein dargestelltes Bild sieht von jeder Richtung gleich aus.

Zu den Nachteilen dieser Technologie gehört zum Einen der hohe Preis, weshalb sie vorwiegend in kleinen portablen Geräten wie Smartphones oder Tablets zu finden sind. Zum Anderen leiden sie unter einer begrenzten Lebensdauer: Insbesondere bei organischen LED-Bildschirmen (AMOLED) degenieren die Pixel in Abhängigkeit der Nutzungsdauer und abgestrahlter Lichtleistung und ihre Lichtleistung reduziert sich. Abbildung 2.5 zeigt diesen Effekt, wie er bei der radiometrischen Vermessung eines AMOLED-Bildschirms beobachtet werden konnte.

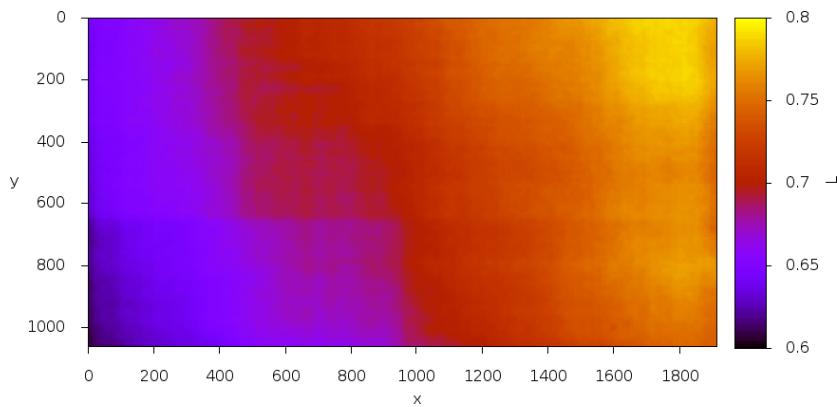


Abbildung 2.5.: Die senkrecht zur Oberfläche gemessene, relative Strahldichte bei einem Samsung S4 Smartphone Bildschirm (AMOLED-Technologie). Man kann deutlich erkennen, dass auf der linken Seite, insbesondere im unteren Bereich, die Lichtleistung reduziert ist. Dort wurden über einen längeren Zeitraum weiße, rechteckige Flächen mit maximaler Intensität angezeigt, wodurch sich die Pixel “abgenutzt” haben.

2.5. Kameras

Eine Kamera sammelt Lichtstrahlen ein und bildet sie auf eine Ebene ab. Bei einer *idealen Kamera* geschieht dies durch eine punktförmige Blende, durch die das Licht fällt und auf einen planen Sensor abgebildet wird (Abbildung 2.6). Durch die unendlich kleine Blende, die dabei angenommen wird, werden vom Sensor im Grunde genommen einzelne Lichtstrahlen wahrgenommen - der Sensor misst also die Strahldichte, die durch die Punktblende verläuft. Dieses Modell wird auch als *Pinhole-Modell* bezeichnet, und kann zur geometrischen Modellierung von echten Kameras verwendet werden. Der Sensor der Kamera besitzt dabei ein gewisses Ansprechverhalten das beschreibt, wie die aufgenommene Strahldichte in einen digitalen Ausgabewert umgesetzt wird.

Echte Kameras verwenden natürlich keine punktförmige Blende, sondern nutzen ein Linsensystem. Es erfüllt dabei allerdings die selbe Funktion, weshalb auch sie mit dem Pinhole-Modell beschrieben werden können. Im Unterschied zur idealen Kamera treten bei einer echten Kamera weitere, meist unerwünschte Effekte auf, wie beispielsweise Verzerrungen durch die Linsen. Sie müssen separat betrachtet und korrigiert werden, und werden vom Pinhole-Modell nicht erfasst.

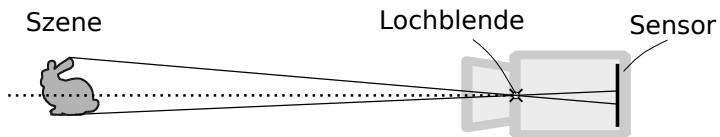


Abbildung 2.6.: Die ideale Kamera besitzt eine punktförmige Blende, durch welche die Lichtstrahlen einer Szene auf eine Sensorebene abgebildet wird. Dieser Prozess kann mit einer perspektivischen Projektion modelliert werden.

Mathematisch lässt sich die Abbildung, die in einer idealen Kamera geschieht, als eine Zentralperspektivische Projektion (engl “perspective projection”) mit einer 3×3 Matrix beschreiben [06]. Diese Matrix C wird als auch als die *intrinsischen Parameter* der Kamera bezeichnet, und beschreibt, wie das Licht aus der Szene auf die einzelnen Sensorpixel abgebildet wird. Die Rotation (3×3 -Matrix R) und Translation (Vektor \vec{t}) der Kamera in der Welt werden mit einer 3×4 Transformationsmatrix $D = [R|\vec{t}]$ erfasst, welche als die *extrinsischen Parameter* bezeichnet wird.

$$s \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = C[R|\vec{t}] \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = \begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \quad (2.3)$$

Die Matrix C bildet einen Punkt der Szene, welcher die Koordinaten (x, y, z) besitzt, auf ein Sensorpixel mit den Koordinaten (u, v) ab. Sie werden dabei mit homogenen Koordinaten beschrieben. C beinhaltet die geometrischen Eigenschaften der Kamera: f_x und f_y ist die Brennweite, ausgedrückt in Pixelkoordinaten, c_x und c_y ist der Punkt auf dem Sensor, durch den die optische Achse verläuft.

Kennt man die intrinsischen Parameter einer Kamera, so bezeichnet man sie als *geometrisch kalibriert*. Damit ist es unter anderem möglich, anhand eines Eingabebildes die extrinsischen Parameter zu berechnen - eine kalibrierte Kamera kann also zur Positionsbestimmung eingesetzt werden.

2.5.1. Positionsberechnung mit Kameras

Anhand der Kameramatrix kann die Richtung des Lichtstrahls berechnet werden, der auf ein bestimmtes Sensorpixel fällt. Platziert man ein Muster mit einer bekannten Geometrie im Frustum der Kamera, so kann man es in der Aufnahme lokalisieren und anhand der Pixelpositionen die extrinsischen Parameter berechnen. Kennt man die Position der Muster in der Welt, so ist auch die Rotation und Position der Kamera innerhalb der Welt berechenbar.

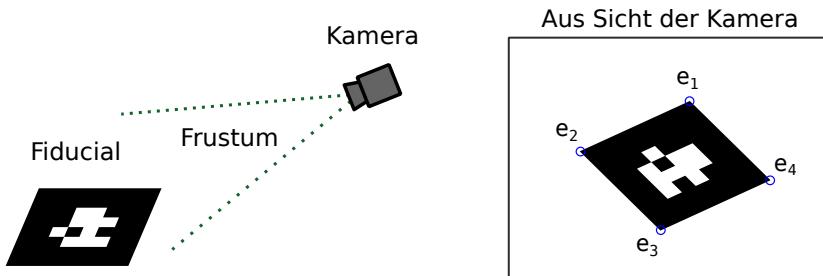


Abbildung 2.7.: Positionsbestimmung mit einer geometrisch kalibrierten Kamera: Im Frustum der Kamera befindet sich ein Muster (ein “Fiducial”), dessen Geometrie und Lage im Raum bekannt ist. Es wird im Kamerabild lokalisiert, und die Position der Eckpunkte e_1-e_4 berechnet. Durch Anwendung des Pinhole-Modells (Gleichung 2.3) kann damit ein Gleichungssystem aufgestellt werden, mit dem nach den extrinsischen Parametern gelöst werden kann.

Diese Verfahren wird gerne im Bereich Augmented Reality (“AR”) verwendet: Kennt man nämlich die extrinsischen Parameter der Kamera, so lässt sich damit ganz einfach ein dreidimensionales Objekt aus dem richtigen Blickwinkel rendern und in den Kamerastream überlagern.

Eine freie Implementierung eines solchen AR-Systems ist die ARToolKit Library [08]. Sie setzt simple schwarze Quadrate als Fiducials ein, die ganz einfach auf Papier ausgedruckt werden können. In der Mitte der Fiducials befindet sich ein Muster, durch das sie sich im Kamerastream auseinanderhalten und eindeutig identifizieren lassen. Damit ist es möglich, nahezu beliebig viele Fiducials gleichzeitig zu verwenden. In dieser Arbeit wird ARToolKit zur Positionsbestimmung verwendet, da nicht nur sehr schnell ist sondern auch zuverlässig mit einfachen Webcams funktioniert und viele Fiducials gleichzeitig eingesetzt werden können.

3. Verwandte Arbeiten

Das System das in dieser Arbeit vorgestellt wird ist eine Light-Stage (siehe Kapitel 2.2.2). Im Gegensatz zu den üblichen Light-Stage-Konstruktionen werden die Lichtquellen jedoch nicht mit diskreten Lampen realisiert, sondern mit einem mobilen Bildschirm. Dadurch kann zwar zu jedem Zeitpunkt immer nur ein Teil der einfallenden Beleuchtung erzeugt werden, dafür aber mit einer extrem hohen Auflösung. Angenommen, die Fläche einer Light-Stage-Lampe beträgt 7cm^2 und der Bildschirm hat eine Pixeldichte von 96 DPI. Verwendet man einen Bildschirm, so lassen sich auf der selben Fläche 10^5 Pixel unterbringen. Durch die sehr hohe Auflösung ergeben sich neue Anwendungsfelder, bei denen herkömmliche Light-Stages nicht eingesetzt werden können. Die Zusammenhänge zu den unterschiedlichen Light-Stage-Konstruktionen werden in Abschnitt 3.1 behandelt.

Diese Arbeit hat auch Ähnlichkeiten mit den *Environment-Matting*-Verfahren. Sie ermöglichen es, ein reales Objekt derart von dem Hintergrund (engl: “Backdrop”) zu separieren, dass es realistisch vor einem beliebigen anderen Hintergrund gesetzt werden kann. Dazu wird der Lichtfluss zwischen Hintergrund, Objekt und Kamera modelliert und rekonstruiert, sodass beispielsweise auch die Lichtbrechung in transparenten Materialien erfasst werden kann, welche das Erscheinungsbild maßgeblich beeinflusst. Environment-Matting kann als Ergänzung zu Light-Stages gesehen werden: Mit diskreten Lampen kann nämlich kein Backdrop erzeugt werden. Die Zusammenhänge zwischen dieser Arbeit und den Environment-Matting-Verfahren werden in 3.2 erklärt.

3.1. Light-Stages

Fotografiert man eine Szene mit einer Kamera, so misst man im Endeffekt die Strahldichte die von vielen Punkten der Szene ausgeht. Wenn man die Szene dabei von unterschiedlichen Richtungen beleuchtet, so kann man das sogenannte Reflektanzfeld rekonstruieren. Aufgrund der additiven Eigenschaften des Lichttransports kann man die Aufnahmen linear kombinieren und die Szene unter einer neuen Beleuchtung “rendern”. Dieses bildbasierte Renderingverfahren nennt sich “Image-Based Relighting”, im Folgenden mit “IBR” abgekürzt. IBR hat die schöne intrinsische Eigenschaft, dass in den gerenderten Bildern so gut wie alle physikalischen Lichteffekte die in einem Material auftreten korrekt wiedergegeben werden können. Bei der Aufnahme von Reflektanzfeldern verwendet man Light-Stages, die sich auf ganz unterschiedliche Arten konstruieren lassen.

Ein großer wissenschaftlicher Beitrag im Bereich IBL und Light-Stages wurde von Debevec und Hawkins geleistet. Sie waren die Ersten die das Reflektanzfeld eines menschlichen Gesichts vermessen, und unter einer neuen Beleuchtung gerendert haben [DHT⁺00]. Sie haben dazu eine Light-Stage entwickelt, die nur eine einzige Lichtquelle verwendet. Die Lampe sitzt auf einem beweglichen Ausleger der rund um die Szene rotiert werden kann und dessen Position zu jedem Zeitpunkt bekannt

3. Verwandte Arbeiten

ist. Ganz ähnlich funktioniert die Free-Form Light-Stage von Masselus et al. [MDA02]. Sie besitzt ebenfalls nur eine Lichtquelle, die dabei jedoch von Hand gehalten und um die Szene herum bewegt wird. Ihre Position kann dann anhand der Lichtreflexion an einer Kugeloberfläche berechnet werden. Das gleiche Prinzip wird auch bei Fuchs et al. [FBS05] angewendet. Sie erzeugen die einfallende Beleuchtung allerdings indirekt, über die diffuse Reflexion an den weißen Wänden eines Raumes. In der Arbeit von Mohan et al. [MTB⁺05] wird das Licht stattdessen mit einem computergesteuerten Scheinwerfer erzeugt, der eine diffuse weiße Kiste von innen anstrahlt. Anhand der Geometrie und der Scheinwerferausrichtung ist die Position der diffusen Lichtquelle zu jedem Zeitpunkt bekannt.

Da für IBR die einfallende Beleuchtung zu jedem Zeitpunkt nur aus *einer* Richtung kommen muss, kann sie auch mit einem mobilen Bildschirm erzeugt werden. Das Verfahren, das in dieser Arbeit vorgestellt wird, kann hier also verwendet werden.

Es gibt jedoch auch Anwendungen in denen eine Szene aus vielen Richtungen *gleichzeitig* beleuchtet werden muss, beispielsweise wenn die Beleuchtung in Echtzeit erzeugt werden soll. Hier sind insbesondere die Arbeiten von Debevec et al. [DWT⁺02] und Lamond et al. [LCE⁺06] zu nennen. Die Light-Stages, die sie dabei verwenden, bestehen aus zahlreichen computergesteuerten, farbigen Lichtquellen die (halb-) kugelförmig um die Szene angeordnet sind. Mit ihnen kann eine einfallende Beleuchtung in Echtzeit erzeugt werden. Dadurch ist es möglich, eine Person unter einer zeitvarianten Beleuchtung zu filmen, weshalb das Prinzip in der Filmproduktion eingesetzt wird. Das Verfahren von Anrys et al. [ADW04] verwendet ebenfalls eine Light-Stage mit vielen Lichtquellen, allerdings um ein Objekt statisch, anhand einer Benutzervorgabe, zu beleuchten. Auch hier muss das Licht aus vielen Richtungen gleichzeitig in die Szene einfallen.

Da mit einem mobilen Bildschirm immer nur ein Teil der einfallenden Beleuchtung erzeugt werden kann, lässt er sich für Echtzeitanwendungen nicht einsetzen.

Durch die sehr hohe Auflösung eines Bildschirms ergeben sich allerdings auch neue Anwendungen, bei denen die üblichen Light-Stage-Konstruktionen nicht verwendet werden können. Ein spiegelndes Objekt lässt sich nämlich nicht mit diskreten Lampen beleuchten, da sie in der Oberfläche sichtbar werden. Verwendet man jedoch einen Bildschirm als Lichtquelle, so ist die Auflösung der Beleuchtung so hoch, dass die einzelnen Lichtquellen nicht mehr auf der Oberfläche auszumachen sind - es entsteht eine zusammenhängende Lichtfläche. Mit einer hohen Auflösung kann man also eine Klasse von Objekten beleuchten, bei der diskrete Lampen versagen.

3.2. Environment-Matting

Environment-Matting kann man als Weiterentwicklung der normalen Matting-Verfahren ansehen. Bei dem einfachsten Matting-Verfahren, dem Bluescreen-Matting [SB96], wird der Hintergrund anhand des Farbtone vom Vordergrund getrennt. Das funktioniert allerdings nur, wenn sich die Szene farblich deutlich von der Hintergrundfarbe unterscheidet. Mit transparenten oder spiegelnden Objekten ist Matting deshalb nicht möglich.

Das Environment-Matting erweitert das Matting-Prinzip, indem es den Lichtfluss zwischen Hintergrund, Szene und Kamera beachtet. Damit ist es beispielsweise möglich, ein Glasobjekt, das komplexe

3.2. Environment-Matting

Lichtbrechungen aufweist, vor einen neuen Hintergrund zu platzieren. Auch Szenen mit feinen Strukturen wie Haare und Fell lassen sich so deutlich besser von einem Hintergrund trennen. Das Verfahren wurde erstmals von Zongker [ZWCS99] vorgestellt, der auch den Begriff “Environment-Matting” geprägt hat, und konnte im Laufe der Zeit verbessert werden [CZH⁺00, WFZ02].

Environment-Matting stellen eine gute Ergänzung zu IBR dar: Mit einer Light-Stage lässt sich nämlich kein Hintergrund erzeugen. Eine Kombination beider Prinzipien wird in den Arbeiten von Matusik et al. [MPN⁺02] und Debevec et al. [DWT⁺02] beschrieben.

Environment-Matting kann durch das Verfahren, das im weiteren Verlauf beschrieben wird, zwar nicht ersetzt werden, das Endergebniss ist aber vergleichbar: Durch die hohe Auflösung werden transparente und spiegelnde Objekte korrekt beleuchtet und ein Backdrop wird dabei automatisch erzeugt. Möchte man die Beleuchtung ändern, so muss man sie allerdings auch neu herstellen und aufnehmen, und kann sie nicht einfach neu berechnen.

4. Modell und Kalibrierung

Bildschirme und Kameras sind computergesteuerte Geräte die mit der realen Welt in Form von Licht interagieren. Ein Bildschirm setzt dabei seine Eingaben, das sind die Werte im *Framebuffer* der Grafikkarte, in eine gewisse Strahldichteverteilung um. Eine Kamera nimmt hingegen die in einem Punkt einfallende Strahldichte auf und digitalisiert sie wieder. Die einzelnen Pixel können dabei in beiden Fällen unabhängig angesteuert und ausgelesen werden.

Die Zusammenhänge zwischen den digitalen und den physikalischen Werten müssen mit geeigneten Modellen beschrieben werden. Modelliert und rekonstruiert man die Funktionsweise der Hardware ausreichend genau, so kann man Vorhersagen über ihr Verhalten treffen. Damit ist es zum Beispiel möglich mit dem Bildschirm eine vorgegebene Strahldichteverteilung in einem Punkt zu erzeugen - man kann ihn also zum Herstellen einer einfallenden Beleuchtung verwenden.

Neben den einzelnen Komponenten muss man auch ihre Anordnung im Gesamtsystem betrachten. Die Beziehungen, die zwischen den Komponenten herrschen, müssen ebenfalls mit einem Modell erfasst werden. Für eine komplette Beschreibung müssen alle Modelle *kalibriert* werden, wobei Messungen an der Hardware durchzuführen sind.

Es soll nun zuerst die Gesamtanordnung aller Komponenten in Abbildung 4.1 betrachtet werden. Das System beinhaltet einen mobilen Bildschirm der zur Szene hin ausgerichtet ist und einen festen Abstand r_l zum Ursprung O besitzt - der Benutzer bewegt ihn wie auf einer Kugeloberfläche rund um die Szene. Es gibt zwei Kameras: Eine am Bildschirm befestigte Webcam dient der Positionsbestimmung, und eine fixe DSLR-Kamera fotografiert die Szene während der Beleuchtung. Die Positionsberechnung der Webcam geschieht anhand von Fiducials, die so in der Welt angeordnet sind dass sie sich außerhalb der Szene befinden, aber von der Webcam gesehen werden können. Anhand der Webcamposition kann dann die Position des Bildschirms berechnen werden, womit sich letztendlich auch die Pixel im Raum lokalisieren lassen.

Zur Durchführung dieser Arbeit wurde ein 13 Zoll Laptop mit einem TN-Bildschirm verwendet. Als Webcam kam die im Bildschirm eingebaute Laptop-Kamera zum Einsatz. Die genauen technischen Daten des Laptops und der verwendeten DSLR-Kamera sind im Appendix A.1 zu finden.

Die Modellierung des Systems erfolgt in zwei Teilen: Das **geometrischen Modell** erfasst die räumlichen Beziehungen zwischen den Komponenten und ermöglicht es die Position der Bildschirmpixel im Raum zu berechnen. Mit dem **radiometrische Modell** wird hingegen beschrieben, wie der Bildschirm eine Beleuchtung erzeugt, und wie die Kamera das Licht wieder aufnimmt. Im Folgenden wird nun zuerst das geometrische, und dann das radiometrische Modell vorgestellt und jeweils der Kalibrierungsvorgang erklärt. Dabei wird unter anderem auch gezeigt wie das Gesamtsystem, ohne den Einsatz zusätzlicher Hardware, radiometrisch kalibriert werden kann.

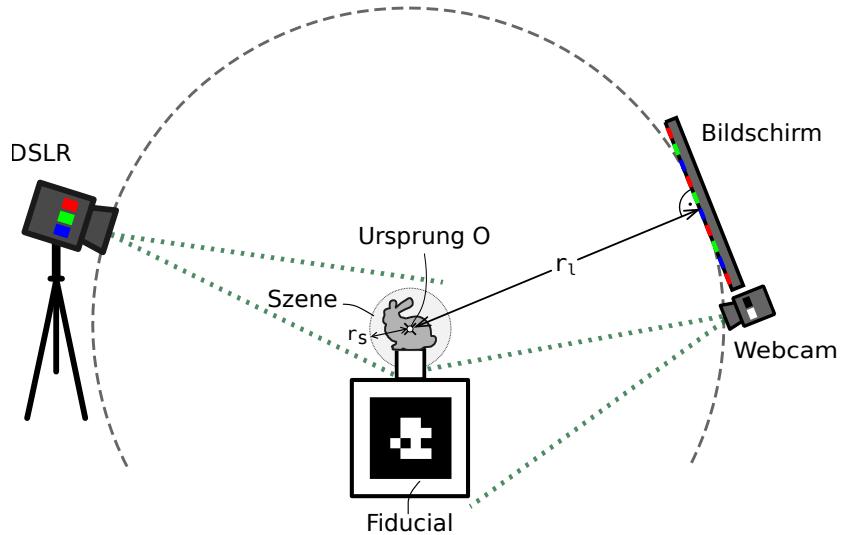


Abbildung 4.1.: Anordnung der Komponenten im System: Die zu beleuchtende Szene befindet sich im Weltursprung O und wird in einer abgedunkelten Umgebung von einem Bildschirm beleuchtet. Die Strecke zwischen Ursprung und Bildschirmzentrum hat die Länge r_l , und der Szenenradius beträgt r_s . Eine am Bildschirm befestigte Webcam nimmt Bilder von geometrischen Markierungen (*Fiducials*) auf, welche sich außerhalb der Szene befinden. Anhand der Webcambilder kann die Kameraposition, und damit auch die Bildschirmposition, berechnet werden. Mit einer feststehenden DSLR-Kamera werden dabei Fotos von der beleuchteten Szene gemacht.

4.1. Das geometrische Modell

Der geometrische Zusammenhang der Komponenten ist in Abbildung 4.2 dargestellt. Das Ziel des geometrischen Modells ist es, die räumliche Position der Bildschirmpixel in Abhängigkeit von der Webcambewegung zu beschreiben.

Zur Berechnung der Kameraposition wird in dieser Arbeit die Bibliothek ARToolKit [08] eingesetzt. Dazu muss die Geometrie der Fiducials im vorhinein bekannt sein, und der Software übergeben werden. Da sie im Voraus entworfen und von Hand konstruiert werden müssen, kann ihre Geometrie als bekannt angesehen werden. Für eine Positionsrekonstruktion müssen des Weiteren die intrinsischen Kameraparameter rekonstruiert werden (siehe Abschnitt 4.1.1). Im weiteren Verlauf wird nun davon ausgegangen, dass die Fiducial-Konfiguration und die Kameraparameter bekannt sind, und mit ARToolKit die extrinsischen Parameter (Position und Rotation) der Webcam berechnet werden können.

Da die Webcam fest am Bildschirm angebracht ist, lässt sich anhand der extrinsischen Parameter auch die Ausrichtung der Bildschirmebene, und damit die Pixelpositionen im Raum berechnen.

Die Translation und Rotation, die zwischen der Kamera und der Bildschirmebene besteht, muss hierzu bekannt sein, kann jedoch im allgemeinen Fall nicht von Hand bestimmt werden. Bei dem eingesetzten

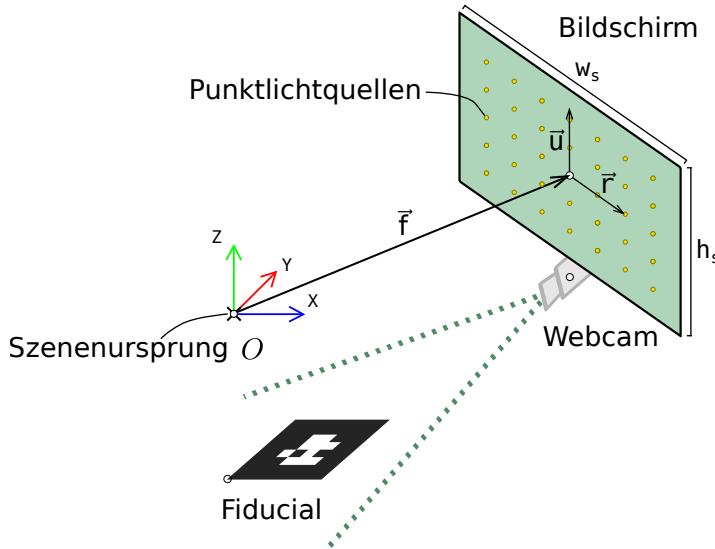


Abbildung 4.2.: Das Geometrisches Modell: Der Mittelpunkt der Szene legt den Ursprung O des Weltkoordinatensystems fest. Der Bildschirm mit den physikalischen Dimensionen w_s und h_s wird als eine Ebene modelliert, die durch die Vektor \vec{f} , \vec{u} und \vec{r} aufgespannt ist, wobei \vec{f} die Bildschirmmitte festlegt. Die Pixel des Bildschirms werden als Punktlichtquellen modelliert die gleichverteilt auf der Ebene liegen. Die Position der Webcam ist dabei relativ zum Bildschirmzentrum definiert und die der Fiducials relativ zum Weltursprung.

Laptop ist die Webcam zudem noch direkt im Rahmen des Bildschirms eingebaut. Maße lassen sich also nicht ohne Weiteres von Hand abnehmen.

In der Praxis wurde jedoch festgestellt, dass sich das Problem dadurch sogar vereinfacht: Aufgrund der flachen Bauweise des Laptopbildschirms befindet sich die Blendenöffnung ungefähr in der Bildschirmebene. Beim Vermessen des Frustums der Laptopkamera konnte zudem herausgefunden werden, dass die optische Achse parallel zur Bildschirmnormalen ist. In diesem Fall ist die Sensorebene parallel zur Bildschirmebene, und die Vektoren \vec{u} und \vec{r} gehen direkt aus den extrinsischen Parametern hervor. Zum Kalibrieren muss also lediglich die Translation der Kamera auf der Bildschirmebene gemessen werden, was leicht mit einem Lineal möglich ist.

4.1.1. Kamera-Parameter

Beide Kameras müssen geometrisch kalibriert werden. Zum Einen werden die intrinsischen Parameter der Webcam für die Positions berechnung benötigt, und zum Anderen müssen die Parameter der DSLR-Kamera bekannt sein, da sie in Abschnitt 4.2.2 zur radiometrischen Kalibrierung des Bildschirms benötigt werden.

Für beide Kameras wird das Pinhole Modell (Abschnitt 2.5) verwendet. Die intrinsischen Parameter C kann man sehr einfach mit der OpenCV Bibliothek [05] berechnen. Die dabei verwendete Kalibrationsroutine basiert auf dem Algorithmus von Zhang et al. [Zha00]. Sie berechnet eine approximierte

4. Modell und Kalibrierung

Kameramatrix anhand einem Satz Kalibrierungsbilder, die ein Schachbrett an unterschiedlichen Stellen im Kamerafrustum zeigen.

Da es sich bei einer echten Kamera nicht um eine Lochkamera mit punktförmiger Blende handelt sondern ein Linsensystem zum Einsatz kommt, treten Verzerrungen im Bild auf. Dadurch werden gerade Linien gekrümmmt auf der Sensorebene abgebildet. Dieser Effekt ist bei der DSLR-Kamera aufgrund der komplizierten Linsenanordnung stärker ausgeprägt als bei der Webcam, deren Objektiv sehr simpel konstruiert ist. Damit eine genaue Positionsbestimmung möglich ist, muss diese Verzerrung auf jeden Fall korrigiert werden. Auch hierfür wird von OpenCV eine geeignete Routine zur Verfügung gestellt, mit der sich radiale und tangentiale Verzerrungen beseitigen lassen.

Mit einer geometrisch kalibrierten Kamera ist es dann möglich, die Lichtstrahlen aufzunehmen, die in einem Punkt einfallen. Dieser Punkt ist durch die Position der Blende im angenommenen Pinhole-Modell gegeben. Im den folgenden Abschnitten wird nun das radiometrische Modell behandelt, mit dem es dann möglich ist auch die Intensität dieser Lichtstrahlen in Form von Strahldichte zu messen.

4.2. Das radiometrische Modell

Das radiometrische Modell beschreibt, wie der Bildschirm digitale Eingabedaten in Strahldichte umsetzt, und wie die Kamera sie wieder digitalisiert. Die Farbkanäle werden dabei im weiteren Verlauf als unabhängig betrachtet: Ein rotes Kamerapixel nimmt also nur das Licht der roten Bildschirmpixel wahr. Alle hier eingeführten Definitionen und Symbole beziehen sich deshalb immer nur auf *einen* Kanal, und sind auf alle drei Kanäle gleichermaßen anzuwenden. Der Begriff "Pixel" wird in diesem Abschnitt stellvertretend für ein einzelnes Subpixel verwendet.

Zur Erklärung des Modells wird nun zunächst der Lichtfluss zwischen einem Pixel des Bildschirms und einem Pixel des Kamerasensors in Abbildung 4.3 betrachtet. Das Licht des Bildschirms kann den Sensor entweder indirekt, über eine Reflexion an der Szene, oder aber auch direkt erreichen. Das direkte Licht wird nicht als Teil der einfallenden Beleuchtung angesehen, im Folgenden wird darum bei der Modellierung ausschließlich die Strahldichte im Ursprung betrachtet.

Damit eine praxistaugliche Modellierung möglich ist, werden mehrere Annahmen getroffen:

Die Beleuchtung ist distant

Das Verhältnis von Lichtquellenabstand r_l zu Szenenradius r_s ist groß genug und erfüllt die Annahme einer distanten Beleuchtung. Damit kann die Szene im weiteren Verlauf als punktförmig angesehen werden.

Pixel sind Punktlichtquellen

Die Fläche eines Pixels wird als unendlich klein angesehen und als Punktlichtquelle modelliert.

r_l und die Ausrichtung des Bildschirms sind festgelegt

Der Abstand r_l zwischen Ursprung und Bildschirmzentrum ist festgelegt und wird immer eingehalten. Der Bildschirm muss außerdem immer so ausgerichtet werden, dass der Vektor \vec{f} orthogonal zur Bildschirmebene steht: die Anzeigefläche ist also immer optimal zur Szene hin ausgerichtet.

Der Bildschirm ist die einzige Lichtquelle

Die Aufnahmen der DSLR-Kamera sollen ausschließlich die Beleuchtung durch den Bildschirm erfassen, und dürfen nicht durch Restlicht aus der Umgebung gestört werden. Das kann in der Praxis durch eine zusätzliche Aufnahme (das sogenannte *Darkframe*) realisiert werden.

Die Szene ist die einzige reflektierende Komponente

Das vom Bildschirm und der Umgebung reflektierte Licht wird im Modell nicht beachtet und als vernachlässigbar klein angesehen.

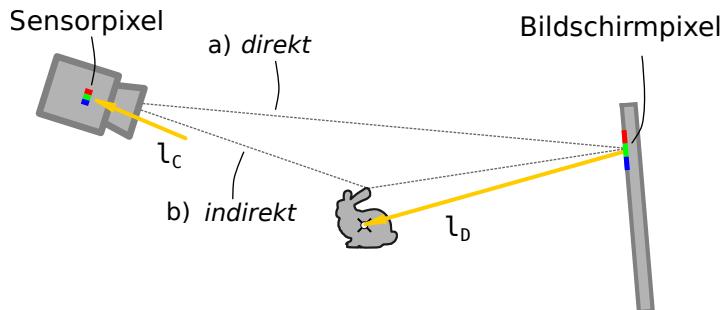


Abbildung 4.3.: Der Lichtfluss zwischen Bildschirm, Szene und Kamera: Ein Lichstrahl mit der Strahldichte l_D wird von einem Pixel emittiert, verläuft durch den Ursprung und erzeugt dort einen Teil der einfallende Beleuchtung. Ein Sensorpixel misst die Strahldichte l_C von einem Lichstrahl, dessen Pfad entweder a) direkt oder b) indirekt verlaufen kann.

Mit Hilfe dieser Annahmen kann die Beleuchtung die vom Bildschirm erzeugt wird durch die in den Ursprung einfallende Strahldichte beschrieben werden. Sie wird in dieser Arbeit mit dem Symbol L bezeichnet, und als eine einfache Matrix dargestellt, in der die Strahldichte l_D jedes Pixels gespeichert ist.

Der radiometrische Zusammenhang zwischen den Bildschirm- und den Kamerapixeln wird mit einer Verkettung beider Systeme beschrieben. Abbildung 4.4 zeigt ihre Interaktion in Form eines Blockdiagramms. Beide Geräte besitzen ein Ansprechverhalten, das ganz allgemein in Form von nichtlinearen Funktionen beschrieben wird.

Ist das Ansprechverhalten beider Geräte bekannt so kann man es invertieren. Damit ist es möglich von der Ausgabe eines Gerätes auf seine Eingabewerte zu schließen. Beispielsweise kann anhand eines aufgezeichneten RAW-Wertes v_C , auf die Strahldichte l_C geschlossen werden, welche die Kamera aufgenommen hat. Man kann sie also zum Messen von Strahldichte einsetzen. Es kann aber auch anhand einer Strahldichte l_D den zugehörigen Eingabewert v_D berechnen - der Bildschirm kann also zum Erzeugen einer einfallenden Strahldichteverteilung verwendet werden. Damit eine Invertierung möglich ist, müssen die Ansprechkurven stetig sein, was jedoch bei Bildschirmen und Kameras aufgrund ihrer Funktionsweise angenommen werden kann.

Es ist an dieser Stelle anzumerken, dass durch die Annahme der unabhängigen Kanäle das Übersprechen zwischen den Farbkanälen des Bildschirms und den Farbkanälen der Kamera vernachlässigt wird. Durch das breite Frequenzspektrum der verbauten Farbfilter passiert es, dass sich beispielsweise

4. Modell und Kalibrierung

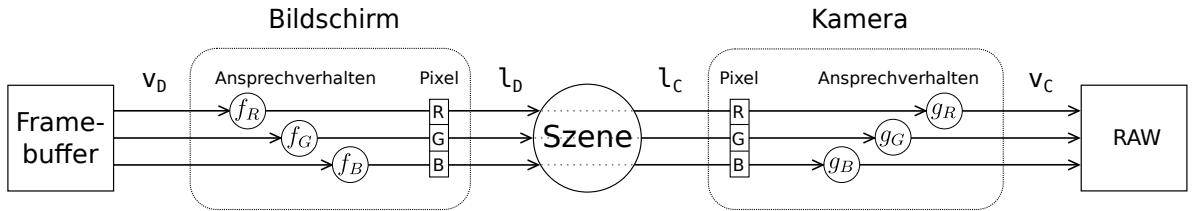


Abbildung 4.4.: Blockdiagramm der radiometrischen Beziehungen: Ein Subpixel des Bildschirms setzt einen Wert v_D aus dem Framebuffer der Grafikkarte in die Strahldichte l_D um. In der Kamera wird von einem Subpixel des Sensors die Strahldichte l_C wahrgenommen, und wieder in einen digitalen Wert v_C umgesetzt. Das Ansprechverhalten beider Geräte ist nichtlinear und wird durch eine Menge aus Funktionen beschrieben (mit f und g angedeutet). Jedes Bildschirmpixel darf dabei ein eigenes Ansprechverhalten besitzen.

das Licht der blauen Bildschirmpixel auch auf den Grünkanal der Kamera auswirkt. Dies führt zu einer verfälschten Farbwiedergabe, kann jedoch beispielsweise durch einen Weißabgleich wieder korrigiert werden [RWPD05].

Im Folgenden wird nun die radiometrische Kalibrierung der DSLR-Kamera beschrieben. Damit kann sie dann im Anschluss zum Messen von Strahldichte eingesetzt werden.

4.2.1. Ansprechverhalten der Kamera

Das Ansprechverhalten einer Kamera beschreibt, wie die aufgenommene Strahldichte durch den Sensor und die Elektronik in einen digitalen Messwert umgewandelt wird. Sind die Ansprechkurven bekannt, so kann man sie invertieren und anhand von den Kamerabildern auf die aufgenommene Strahldichte schließen. Die Kamera kann also als Messgerät eingesetzt werden. Zusammen mit dem Pinhole-Modell ist es dann möglich, die in einem Punkt einfallende Strahldichte zu vermessen, womit es letztendlich auch möglich ist den Bildschirm radiometrisch zu kalibrieren.

Es existieren mehrere Verfahren, mit denen sich das Ansprechverhalten eines Kamera-Sensors ohne großen Aufwand bestimmen lässt. Bei den Methoden von Robertson et al. [RBS99], Mitsunaga et al. [MN99] und Debevec et al. [DM97] wird dazu mit einer fixen Kamera eine Reihe Bilder von einer statischen Szene geschossen, und dabei die Belichtungszeit variiert. Da die vom Sensor aufgenommene relative Strahldichte proportional zur Belichtungszeit ist, kann das Ansprechverhalten rekonstruiert werden.

Das Ansprechverhalten wurde in dieser Arbeit mit dem Verfahren von Robertson [RBS99] bestimmt. Eine freie Implementierung des Robertson-Algorithmus findet sich in der Programmsammlung *pfscalibration* [03]. Die Pixel eines Farbkanals werden dabei als gleichwertig angesehen und es wird eine Ansprechkurve pro Farbkanal rekonstruiert.

In Abbildung 4.5 sind die invertierten Kennlinien von dem Sensor einer Canon EOS 5D Mark II Kamera dargestellt. Betrachtet man die Diagramme, so fällt einem das nahezu lineare Ansprechverhalten ins Auge, das sich über den Großteil des Wertebereiches zieht. In den oberen Werten (>85%), aber auch in

den Schwarzwerten (<1%) verhält sich der Sensor nichtlinear. Die Unterschiede in den Schwarzwerten sieht man am besten in einem Doppellogarithmischen Plot (siehe Abbildung A.1 im Anhang).

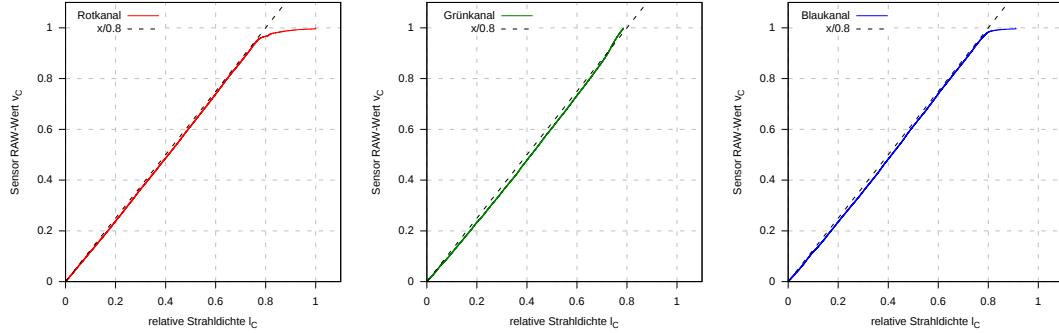


Abbildung 4.5.: Ansprechverhalten von dem Sensors einer Canon EOS 5D Mark II Kamera, rekonstruiert mit dem Programm pfscalibration [03]. Es ist die relative Strahldichte l_C über den RAW-Werten v_C abgetragen, wobei beide Achsen auf den Bereich (0, 1) normiert sind.

Mit einer radiometrisch und geometrisch kalibrierten DSLR-Kamera ist es nun möglich auch den Bildschirm zu kalibrieren.

4.2.2. Ansprechverhalten des Bildschirms

Analog zur Kennlinie einer Kamera besitzt auch ein Bildschirm ein Ansprechverhalten. Mit ihm wird der Zusammenhang zwischen der Strahldichte l_D eines Pixels, und dem jeweiligen Wert v_D im Framebuffer beschrieben. Im weiteren Verlauf wird ein 24-Bit Framebuffer (8 Bit pro Farbkanal) angenommen, jedes Pixel kann also 256 unterschiedliche Werte annehmen.

Das Ansprechverhalten der Pixel lässt sich durch eine Menge Funktionen

$$l_{x,y}(v), \quad v \in [0, 225] \quad (4.1)$$

beschreiben, wobei (x, y) die Pixelkoordinate bezeichnet. Die Zielmenge der Funktionen ist rational, und wird jeweils durch eine minimalen Wert $l_{x,y}^0$, sowie einen maximalen Wert $l_{x,y}^{255}$ beschränkt. Dies entspricht der minimalen und der maximalen Strahldichte, die von einem Pixel ausgeht wenn man den Framebuffer 0 und 255 setzt.

Da $l_{x,y}$ stetig ist kann man das inverse Ansprechverhalten

$$v_{x,y}(l) = (l_{x,y})^{-1}(l) \quad l \in [l_{x,y}^0, l_{x,y}^{255}] \quad (4.2)$$

aller Pixel berechnen. Mit den invertierten Funktionen ist es dann möglich, von einer Strahldichte l_D auf den Framebuffer-Wert v_D zu schließen.

In dieser Arbeit werden alle Ansprechkurven mit stetigen, stückweise linearen Funktionen modelliert. Sie lassen sich einfach invertieren, da sie nur aus Geradenstücken bestehen: Zu jedem gegebenen $l_D \in [l_{x,y}^0, l_{x,y}^{255}]$ kann immer ein v_D berechnet werden, sodass $l_{x,y}(v_D) = l_D$.

4. Modell und Kalibrierung

Die Funktionen $v_{x,y}$ besitzen einen eingeschränkten Definitionsbereich, der nicht für alle Pixel identisch ist. Die Bereichsgrenzen $l_{x,y}^0$ und $l_{x,y}^{255}$ werden deshalb zusätzlich in den Matrizen L_0 und L_{255} gespeichert:

$$\forall(x,y) : \quad L_0(x,y) = l_{x,y}^0 \quad (4.3)$$

$$L_{255}(x,y) = l_{x,y}^{255} \quad (4.4)$$

Im folgenden Abschnitt wird nun erklärt, wie das Ansprechverhalten $l_{x,y}$ eines Bildschirm gemessen werden kann.

Kalibrierungsaufbau

Den Wertebereich aller Ansprechkurven $l_{x,y}(v)$ kann man mit einer radiometrisch und geometrisch kalibrierten DSLR-Kamera messen. Hierzu setzt man den Framebuffer $V(x,y)$ aller Pixel auf einen bestimmten Wert und fotografiert das vom Bildschirm emittierte Licht im Ursprung.

Da das Objektiv einer DSLR-Kamera eine große, komplizierte Linsenanordnung darstellt, kann man sie nicht ohne Weiteres von Hand so ausrichten, dass die Strahldichte im Ursprung gemessen wird. Es ist nicht direkt ersichtlich wo sich der Punkt befindet, durch den alle Lichtstrahlen verlaufen (siehe Kapitel 2.5).

Hier kann das Prinzip der Kamerabasierten Positionsbestimmung angewendet werden: Zeigt man Fiducials auf dem Bildschirm an, so kann mit ARToolKit die Position des Pinholes berechnet werden. Damit ist es möglich, die Kamera so auszurichten, dass sie die Strahldichte exakt im Ursprung misst. Eine Voraussetzung hierfür ist, dass die intrinsischen Parameter der DSLR-Kamera für die gewählte Objektivkonfiguration bekannt sind und auch Linsenverzerrungen korrigiert werden.

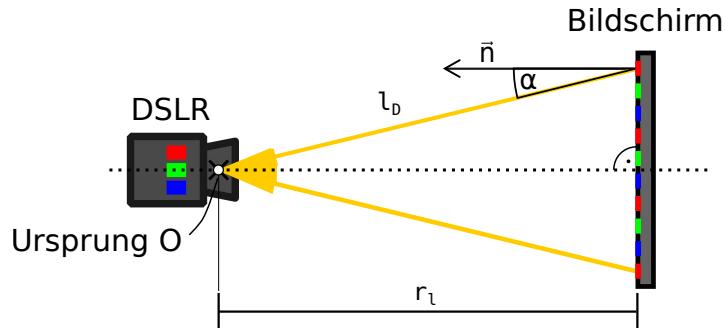


Abbildung 4.6.: Kalibrierungsaufbau: Eine radiometrisch und geometrisch kalibrierte DSLR-Kamera misst die von den Pixeln ausgehende, relative Strahldichte im Ursprung O , welcher sich im Abstand r_l mittig vor dem Bildschirm befindet. Die aufgezeichneten Lichtstrahlen verlassen dabei die Oberfläche unter einem bestimmten Austrittswinkel α , der für jeden Pixel unterschiedlich ist.

Da die drei Farbkanäle als unabhängig angesehen werden, können sie alle gleichzeitig mit nur einer Kameraaufnahme erfasst werden. Ein Überspielen zwischen den Kanälen wird dabei *mit* aufgenommen.

Für die Rekonstruktion von $l_{x,y}$ müssen maximal $2^8 = 256$ Bilder aufgenommen werden. Der Aufwand dieser Prozedur kann durch Unterabtastung und einer geeigneten Interpolation reduziert werden. In dieser Arbeit werden insgesamt 33 Aufnahmen zur Rekonstruktion eingesetzt. Es wird dabei jeder achte Wert, einschließlich den Grenzen $V = 0$ und $V = 255$, vermessen und Zwischenwerte linear interpoliert.

Um eine Zuordnung von den Pixeln der Kalibrierungsbilder zu den Pixeln des Bildschirmes zu ermöglichen wird eine zusätzliche Aufnahme, bei unveränderter Anordnung, geschossen, wobei statt einer Graustufe ein Schachbrett muster angezeigt wird. Anhand der Eckpunkte des Musters kann die Kameraaufnahme dann durch eine perspektivische Projektion in Bildschirmkoordinaten umgerechnet werden. Die Genauigkeit des Messergebnisses hängt dabei von mehreren Faktoren ab, die bei der Kalibrierung unbedingt beachtet werden müssen:

Restlicht

In der Praxis ist es nicht ohne Weiteres möglich einen Raum komplett abzudunkeln, weshalb bei der Kalibrierung auch Restlicht aus der Umgebung mit aufgenommen wird. Dies lässt sich durch eine weitere Aufnahme lösen, bei der die Hintergrundbeleuchtung des Bildschirms ausgeschaltet wird. So kann das Restlicht gezielt gemessen und von den Kalibrierungsaufnahmen abgezogen werden. Diese zweite Aufnahme wird in dieser Arbeit als *Darkframe* bezeichnet.

Bildrauschen

Bei der Aufnahme von Kalibrierungsdaten muss besonders auf ein gutes Signal-Rauschverhältnis geachtet werden. Ist das Rauschen des Kamerasensors zu groß, so können kleinere Lichtmengen nicht mehr genau vermessen werden. Bildrauschen kann man zum Beispiel dadurch reduzieren, indem man Bilder mehrfach aufnimmt und den Durchschnitt bildet.

Bildschirmrand

Bei Messungen am Laptopbildschirm wurde festgestellt, dass die Pixel im Randbereich einen deutlich eingeschränkten Dynamikbereich besitzen: Die Hintergrundbeleuchtung ist dort deutlich unregelmäßiger als in der Mitte des Anzeigefeldes, da sich dort beispielsweise die Lichtquellen befinden. Der Bildschirmrand eignet sich damit nur bedingt zur Beleuchtung und sollte ausgespart werden.

Aliasing

Die Oberfläche eines Flachbildschirms besitzt feine Strukturen die wesentlich kleiner sind als die Abmessungen der Subpixel selbst, wie zum Beispiel die Abstände die sich dazwischen befinden. Durch die hohe räumliche Frequenz kommt es beim Fotografieren zu Aliasing-Effekten. Zur Abhilfe kann hier entweder ein Diffusor vor dem Bildschirm angebracht werden, oder man defokussiert die Kamera minimal. Beides verhält sich im Grunde wie ein Tiefpass, womit hohen Frequenzen herausgefiltert und Aliasing-Effekte eliminiert werden können.

Bei der Kalibrierung wird die Annahme getroffen, dass sich das Ansprechverhalten benachbarter Pixeln nur gering unterscheidet, und über die gesamte Oberfläche betrachtet stetig verläuft. Unter dieser Annahme kann man die räumliche Auflösung der Kalibrierungsdaten reduzieren, wodurch sich auch das Rauschverhältnis verbessert da der Durchschnittswert vieler Pixel zur Berechnung der Ansprechkurven herangezogen werden kann. Dazu wird die genutzte Bildschirmoberfläche in kleine Quadratische Flächenstücke ("Patches") eingeteilt (Abbildung 4.7). Das Ansprechverhalten wird pro

4. Modell und Kalibrierung

Patch berechnet, indem der Durchschnitt der darin enthaltenen Pixel gebildet wird. Damit kann das Sensorrauschen deutlich reduziert werden, da selbst mit sehr kleinen Patches wie beispielsweise $k = 10$, schon 100 Pixel zur Berechnung eines Kurvenstützpunktes verwendet werden.

Die Ansprechkurve eines einzelnen Pixels kann anschließend aus den Kurven benachbarter Patches berechnet werden, indem man sie bilinear interpoliert. Wie das genau funktioniert wird in Abschnitt 6.2 erklärt, da erst zur Laufzeit interpoliert wird.

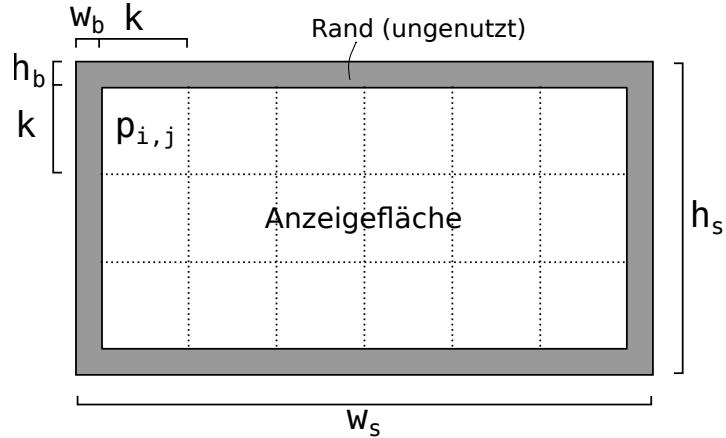


Abbildung 4.7.: Aufteilung der Bildschirmanzeigefläche: Ein schmaler Rand (grau) mit einer Höhe von h_b und einer Breite von w_b Pixeln bleibt wegen einem eingeschränkten Dynamikbereich ungenutzt. Die zur Beleuchtung verwendete Anzeigefläche (weiß) wird in quadratische Patches $p_{i,j}$ der Größe k aufgeteilt (hier übertrieben groß dargestellt). Für jeden Patch wird ein eigenes Ansprechverhalten berechnet, indem der Mittelwert der darin enthaltenen Pixel gebildet wird.

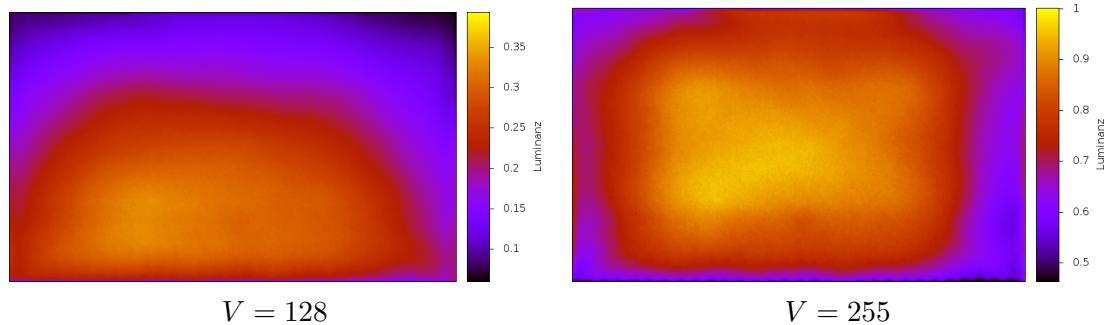


Abbildung 4.8.: Die Heatmaps zeigen die gemessene Strahldichte für die mittlere ($V = 128$) und die maximale Graustufe ($V = 255$) der Kalibrierungsreihe. Die relative Strahldichte wurde nach Gleichung A.3 in Luminanz umgerechnet. Es fällt auf, dass sich die Lichtverteilung in beiden Aufnahmen deutlich unterscheidet - Der Bildschirm besitzt folglich ein, über die Oberfläche betrachtet, inhomogenes Ansprechverhalten.

Das Ansprechverhalten eines LCD-Bildschirms hängt sehr stark von der Richtung ab, mit der das Licht die Oberfläche verlässt. In Abbildung 4.9 ist das rekonstruierte Ansprechverhalten des Laptop-

bildschirms dargestellt. Dabei betrug $r_l = 50 \text{ cm}$ und der Bildschirm wurde in 44×24 Patches der Größe $k = 30$ Pixel aufgeteilt. Die Position eines Graphen entspricht dabei ungefähr der Position des Patches, dessen Ansprechverhalten dargestellt wird: Der Graph links oben zeigt die Ansprechkurven von Patch $p_{0,0}$, welcher sich in der linken oberen Bildschirmecke befindet. Der Graph ganz rechts oben zeigt die des Patches aus der rechten oberen Ecke, und so weiter.

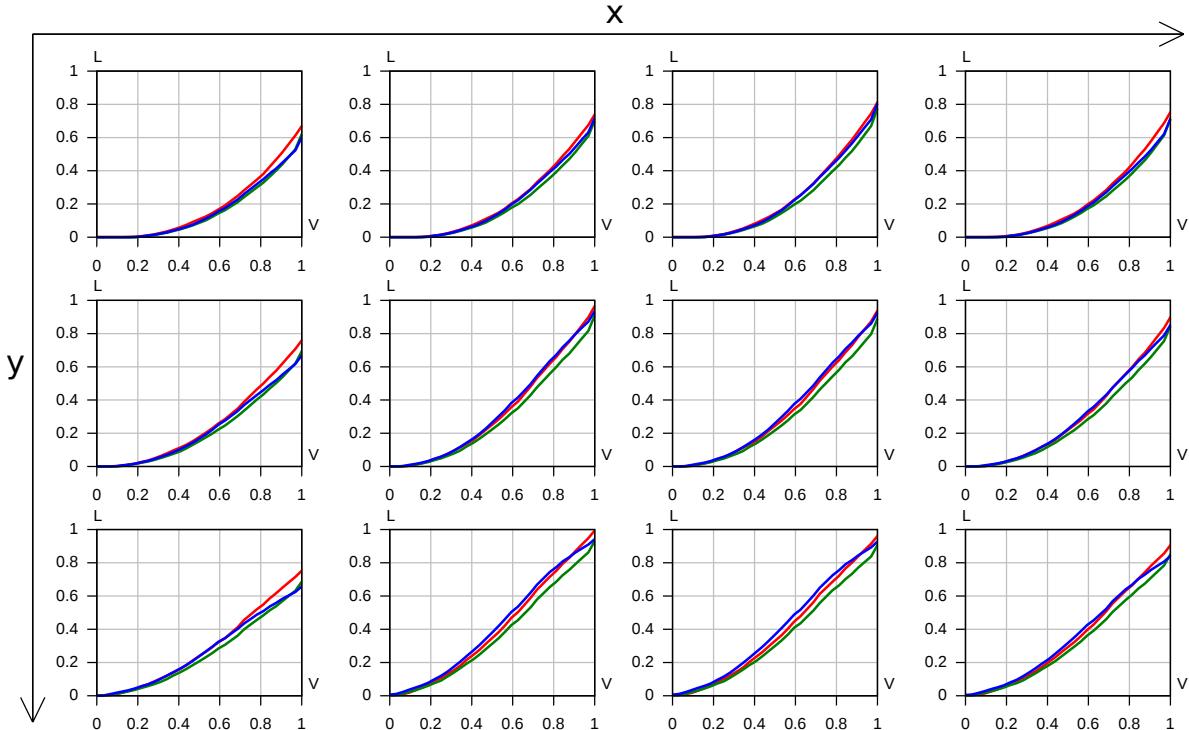


Abbildung 4.9.: Ansprechverhalten des verwendeten TN-Bildschirms: Zur Darstellung wurden 12 Patches gleichmäßig von der Oberfläche ausgewählt. Die Position eines Graphen entspricht der ungefähren Patchposition auf dem Bildschirm. Bei der Aufnahme war $r_l = 50\text{cm}$, und es wurden 44×24 Patches der Größe $k = 30$ Pixel verwendet (etwa 1cm groß). Man kann deutlich erkennen, wie der unterschiedliche Lichtaustrittswinkel das Ansprechverhalten beeinflusst: Mit zunehmendem Blickwinkel reduziert sich nicht nur die maximale Strahldichte, sondern es ändert sich auch das Ansprechverhalten. Der Dynamikbereich ist in der Mitte des Bildschirms darum deutlich größer als im Randbereich.

5. Tracking-Stage

Damit die Position der Webcam berechnet werden kann müssen die Fiducials derart in der Welt angebracht werden, dass sie zwar außerhalb der Szene, aber innerhalb des Kamerafrustums liegen. Wenn der Bildschirm auf einer Halbkugel um die Szene bewegt wird, sollen sich zu jedem Zeitpunkt Fiducials im Blickfeld der Webcam befinden. Die Positionsrechnung soll außerdem in einem dunklen Raum funktionieren, weshalb die Marker beleuchtet werden müssen.

In diesem Kapitel wird eine Aufbau vorgestellt, mit dem das möglich ist. Dieser Aufbau fungiert als eine Art Plattform für die Szene, und wird deshalb als "Tracking-Stage" bezeichnet.

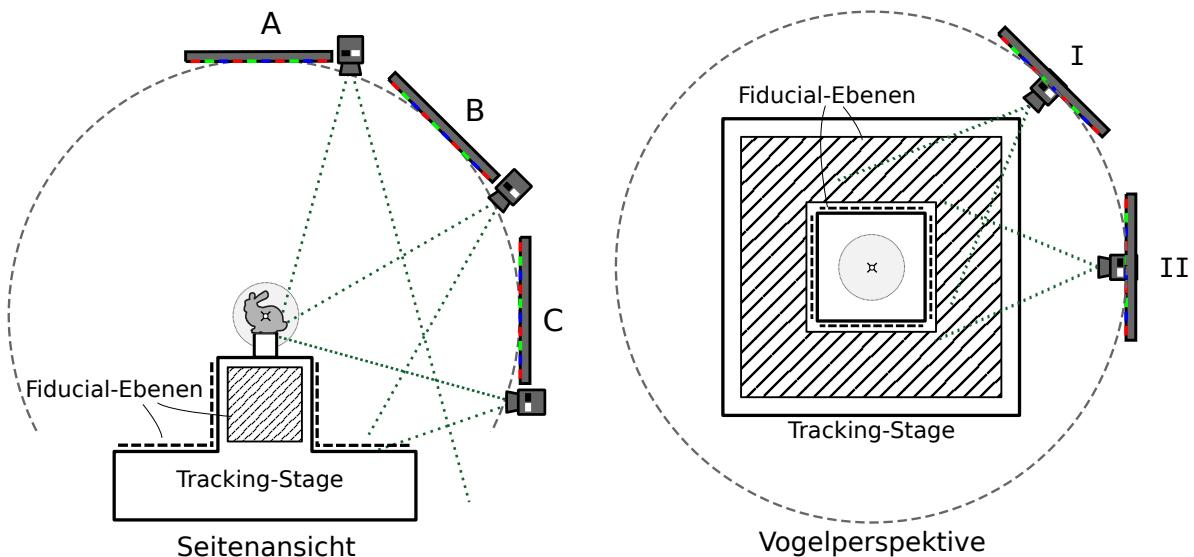


Abbildung 5.1: Die Fiducial-Ebenen der Tracking-Stage, aus der Seitenansicht (links) und der Vogelperspektive (rechts). Die schraffierten und gestrichelten Flächen sind so ausgerichtet, dass sie sich immer im Frustum der Kamera befinden, wenn der Bildschirm auf einer Halbkugel rund um die Szene bewegt wird. Abhängig von der vertikalen (A, B, C) und der horizontalen (I, II) Ausrichtung liegen immer eine, zwei oder drei Ebenen im Frustum der Kamera.

Die Marker sind, wie in Abbildung 5.1) dargestellt, auf mehreren orthogonalen Flächen ("Fiducial-Ebenen") angeordnet. Die genaue Anzahl und Position der Fiducials ist für die geometrische Betrachtung zunächst unwichtig. In der Seitenansicht ist zu erkennen, dass in jeder der Positionen A, B und C mindestens eine der Ebenen im Frustum der Webcam liegt. Die Kamera ist dazu wie dargestellt an der Unterseite des Bildschirms angebracht, da sie so am Besten zu den Ebenen ausgerichtet ist.

5. Tracking-Stage

Der Benutzer kann mit dieser Anordnung den Bildschirm um die Szene herum bewegen, und eine halbkugelförmige Lichtfläche erzeugen. Je nach Öffnungswinkel (Field of View) der Kamera können dabei auch niedrigere Beleuchtungswinkel erreicht werden, sodass mehr als die obere Hälfte der eingehenden Beleuchtung erzeugt werden kann.

5.1. Auswahl der Fiducials

In dieser Arbeit wird ARToolKit zur Positions berechnung verwendet. Das Programm kann mit beliebig vielen Fiducials gleichzeitig umgehen, die dabei beliebig in der Welt angeordnet sein dürfen. Sobald mindestens ein Marker im Kamerabild sichtbar ist, kann auch die Position berechnet werden. Damit die Fiducials dabei identifiziert werden können, müssen sich ihre Muster alle unterscheiden.

Da für die Tracking-Stage sehr viele Marker benötigt werden, macht es Sinn sie automatisch zu generieren. Dazu wurde die Musterfläche der Fiducials in 4x4 Zellen eingeteilt, wobei jede Zelle entweder schwarz oder weiß sein darf. Ein Muster kann so ganz einfach mit einer 16 Bit Ganzzahl dargestellt werden. Eine größere Mustermenge lässt sich dann mit einem Pseudo-Zufallsgenerator erzeugen. Man muss dabei lediglich darauf achten, dass keine rotationssymmetrischen Muster mit in die Menge aufgenommen werden. Da so insgesamt 2^{14} unterschiedliche Muster möglich sind, kann man die Menge auch noch weiter eingeschränken: Man kann beispielsweise erzwingen, dass sich alle Muster in einer Mindestanzahl an Zellen unterscheidet (hier: 6 Zellen), wodurch sie dann beim Tracking besser auseinander gehalten werden können. Abbildung 5.2 zeigt die ersten 24 Fiducials der so generierten Mustermenge.

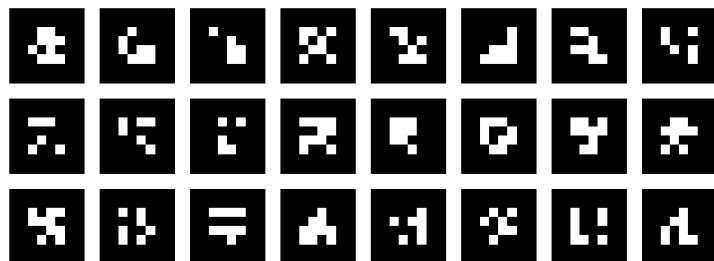


Abbildung 5.2.: Die ersten 24 der algorithmisch generierten Fiducials. Die 16 Zellen werden mit binärzahlen dargestellt, und mit einem Zufallsgenerator erzeugt.

Es ist klar dass ein Fiducial, das im Kamerabild eine größerer Fläche aufweist, robuster gegen Sensorsrauschen ist ein ein kleineres: Je länger die Umrisslinien der Fiducials im Bild sind, desto mehr Pixel sind beim Berechnen der Eckpunkte beteiligt, und desto weniger Einfluss hat das Rauschen einzelner Sensorpixel auf das Ergebnis. Sind die Marker kleiner, so lassen sich allerdings auch deutlich mehr davon auf der selben Fläche unterbringen. Durch berechnen des Durchschnittswerts beim Auswerten der Position erhöht sich die Genauigkeit wieder. Um das Verhalten dieses Tradeoffs zu analysieren wurde eine einfache Testreihe aufgestellt. Dabei wurde untersucht, wie sich ARToolKit bei unterschiedlichen Fiducialkonfigurationen verhält.

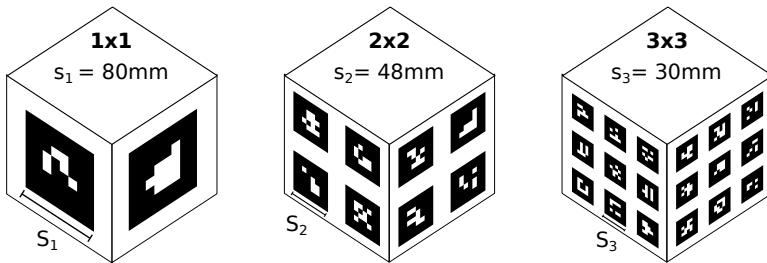


Abbildung 5.3.: Die Fiducial-Testreihe besteht aus drei verschiedenen Konfigurationen, die jeweils aus zwei orthogonalen Ebenen ($15 \times 15\text{ cm}$) bestehen. Darauf befinden sich entweder 1, 4 oder 9 Muster unterschiedlicher Größe. Sie wurden so gewählt dass jede Konfigurationen die gleiche Fläche beansprucht, und die Marker dabei noch von einem weißen Rand umgeben sind, sodass sie auch erkannt werden können.

In der Testreihe werden die drei Musterkonfigurationen 1x1, 2x2 und 3x3 (Abbildung 5.3) untersucht. Jede Konfiguration besteht aus zwei Panels mit der Größe $15 \times 15\text{ cm}$, die im rechten Winkel zueinander angeordnet sind. Die Fiducials wurden mit einem Laserdrucker ausgedruckt und auf den Seiten eines 15cm Holzwürfels aufgeklebt, wobei sich die zwei Panels einer Konfiguration an einer Kante berühren. Der Würfel wurde dann auf eine drehbare Plattform platziert und mit einem Abstand von 50 cm vor der Webcam aufgestellt. Es wurde innerhalb von sechs Sekunden von der Frontalansicht des eine Panels, auf die Frontalansicht des anderen rotiert ($0^\circ - 90^\circ$). In den Positionen 0° , 45° und 90° wurde die Bewegung kurz unterbrochen, um die Auswirkungen von Motion-Blur erfassen zu können.

Abbildung 5.4 zeigt den Abstand zwischen der Kamera und dem Mittelpunkt des Würfels, der von ARToolKit anhand der Marker berechnet wird. Darunter ist jeweils die Anzahl der bei der Berechnung verwendeten Fiducials dargestellt.

Mit der Testreihe konnte gezeigt werden, dass viele kleine Marker einen Vorteil gegenüber wenigen, großen Markern besitzen. Aus diesem Grund werden auf den Fiducial-Ebenen der Tracking-Stage möglichst viele, kleine Marker aufgebracht.

5.2. Konstruktion

Bei der mechanischen Konstruktion der Fiducials ist es von höchster Bedeutung dass enge Toleranzen eingehalten werden. Wenn die tatsächliche Position, Größe oder Orientierung der Fiducials von der Definition abweicht kann ARToolKit die absolute Position nicht korrekt berechnen. Insbesondere hängt das Ergebnis auch davon ab, welche Marker konkret bei der Berechnung verwendet wurden. Da bei der Tracking-Stage zu jedem Zeitpunkt immer nur ein Teil aller Fiducials sichtbar ist, kommt es bei gleichmäßigen Bewegungen zu starken Sprüngen. Beim Entwurf der Tracking-Stage wurde deshalb darauf geachtet, dass die Fiducials mit Toleranzen im Bereich von $\pm 0.5\text{mm}$ im Raum angeordnet werden können.

Die Basis des Aufbaus besteht aus einer $50 \times 60\text{cm}$ großen, 2mm starken Glasplatte aus einem Bilderrahmen, welche die horizontale Fiducial-Ebene darstellt. Die Muster können mit einem gewöhnlichen

5. Tracking-Stage

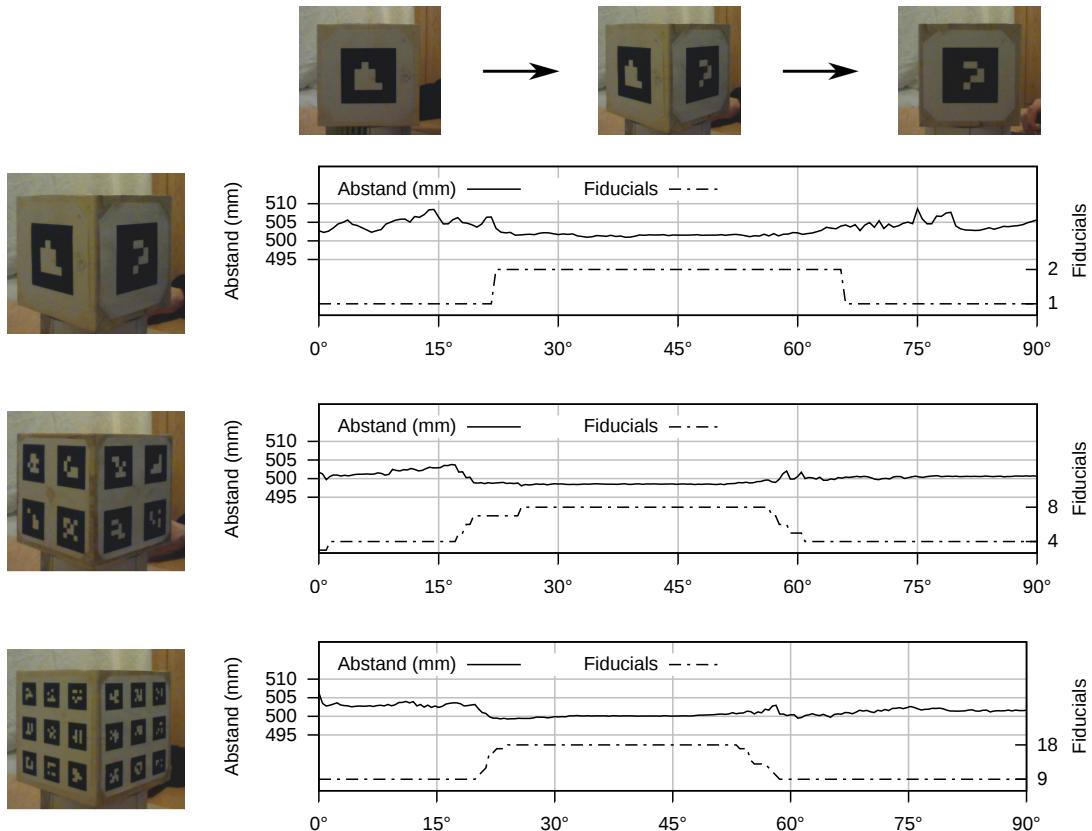


Abbildung 5.4.: Drei Musterkonfigurationen auf einem 15 cm großen Würfel werden mit ARToolKit getrackt. Der Würfel wird dabei langsam um 90 Grad um die senkrechte Achse gedreht. Die Diagramme zeigen den Abstand zum Mittelpunkt des Würfels, sowie die Anzahl der Fiducials die bei der Berechnung verwendet wurden. Bei jeder Konfiguration ist zu sehen, dass der Abstand deutlich genauer berechnet werden kann, sobald Fiducials von beiden orthogonalen Ebenen sichtbar sind. Motion-Blur scheint bei der langsamen Bewegung keinen Einfluss zu haben. Man sieht auch, dass es von Vorteil ist, wenn man viele kleine, anstatt wenige große Fiducials verwendet: Je mehr Marker im Bild sind, desto geringer sind die Auswirkungen des Sensorrauschen.

Laserdrucker auf Papier ausgedruckt werden und ganz einfach auf die Glasplatte geklebt werden. Dabei ist es wichtig dass sie flächig aufgeklebt werden, da sich das Papier bei Luftfeuchtigkeitsänderungen ausdehnen und zusammenziehen kann. Die Glasplatte steht auf mehreren identischen Plastikbechern wird mit zwei LED-Lampen von unten beleuchtet. In der Mitte steht eine Holzkiste (15x15x15 cm), deren Boden und Seitenwände ausgesägt wurden.. Sie ist mit Acrylglasplatten versehen, auf denen die vier vertikalen Fiducial-Ebenen aufgeklebt sind.

Die Fiducials der Tracking-Stage werden vor dem Ausdrucken farblich invertiert, da sie sich so besser zum Durchleuchten eignen. ARToolKit kann ein Fiducial nämlich nur dann erkennen, wenn es von einem weißen Rand umgeben ist. Invertiert man die Muster und den Webcamstream, so

5.3. Evaluation: Tracking-Stage

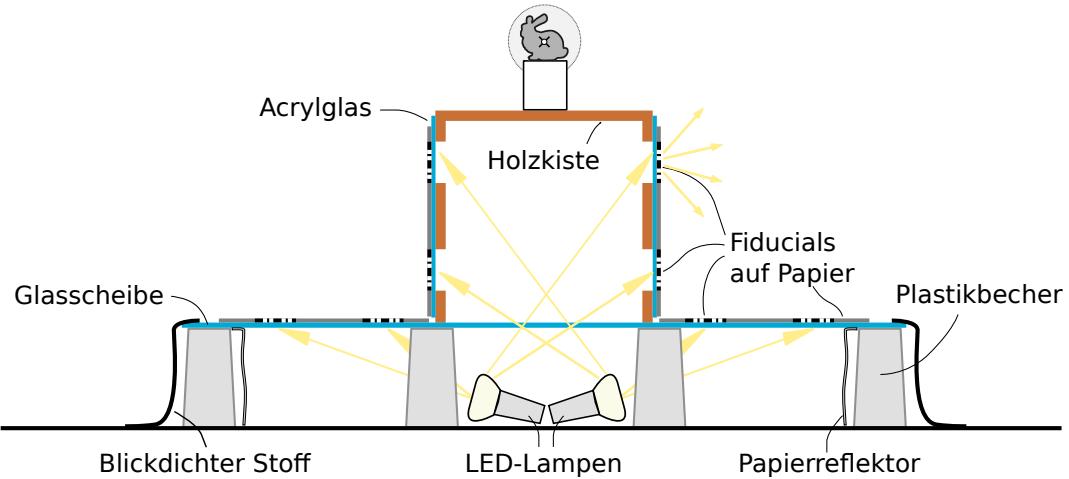


Abbildung 5.5.: Aufbau der Tracking-Stage: Eine Glasplatte, aufgebockt auf Plastikbechern, stellt die Basis dar. Die Fiducials sind auf Papier ausgedruckt und auf die Glasplatte aufgeklebt. In der Mitte befindet sich eine Holzkiste mit ausgesägten Seitenwänden, auf der die vertikalen Fiducials mit Acrylglasplatten befestigt sind. Mit mehreren Lampen wird der ganze Aufbau von innen beleuchtet, Streulicht wird dabei von blickdichtem Stoff abgefangen.

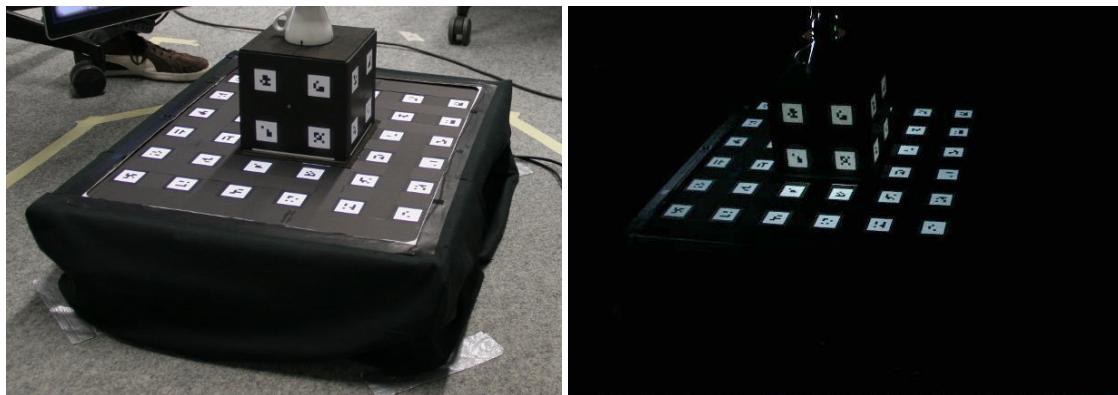


Abbildung 5.6.: Die Tracking-Stage, einmal unter einer Bürobeleuchtung und einmal im dunkeln. Die Lampen sind in beiden Bildern auf die gleiche Helligkeit eingestellt.

muss der weiße Rand nicht beleuchtet werden. Die zu beleuchtende Fiducial-Fläche ist somit kleiner, wodurch weniger Licht in den Raum abgestrahlt wird. Die Fiducial-Ebenen der Tracking-Stage sind in Abbildung A.2 im Anhang dargestellt.

5.3. Evaluation: Tracking-Stage

In diesem Abschnitt wird untersucht, wie gut sich die Tracking-Stage mit ihren beleuchteten Fiducials zur Positionsbestimmung im Dunkeln eignet. Dazu wurde der Laptopbildschirm an drei Positionen

5. Tracking-Stage

bewegt, und so ausgerichtet, wie es bei einer Beleuchtung der Fall ist. Die horizontale und vertikale Ausrichtung des Bildschirms entspricht den Positionen **BI**, **CI** und **CII** aus Abbildung 5.1. In jeder Position wurde das Gerät einmal von Hand gehalten, und einmal stationär aufgestellt, wobei versucht wurde einen Abstand von 50 cm einzuhalten. Es wurden jeweils eine zwei Sekunden lange Videosequenz aufgezeichnet, und mit ARToolKit die Kameraposition berechnet. In den Diagrammen in Abbildung 5.7 ist der so berechnete Abstand zwischen Webcam und dem Weltursprung in Millimeter abgetragen.

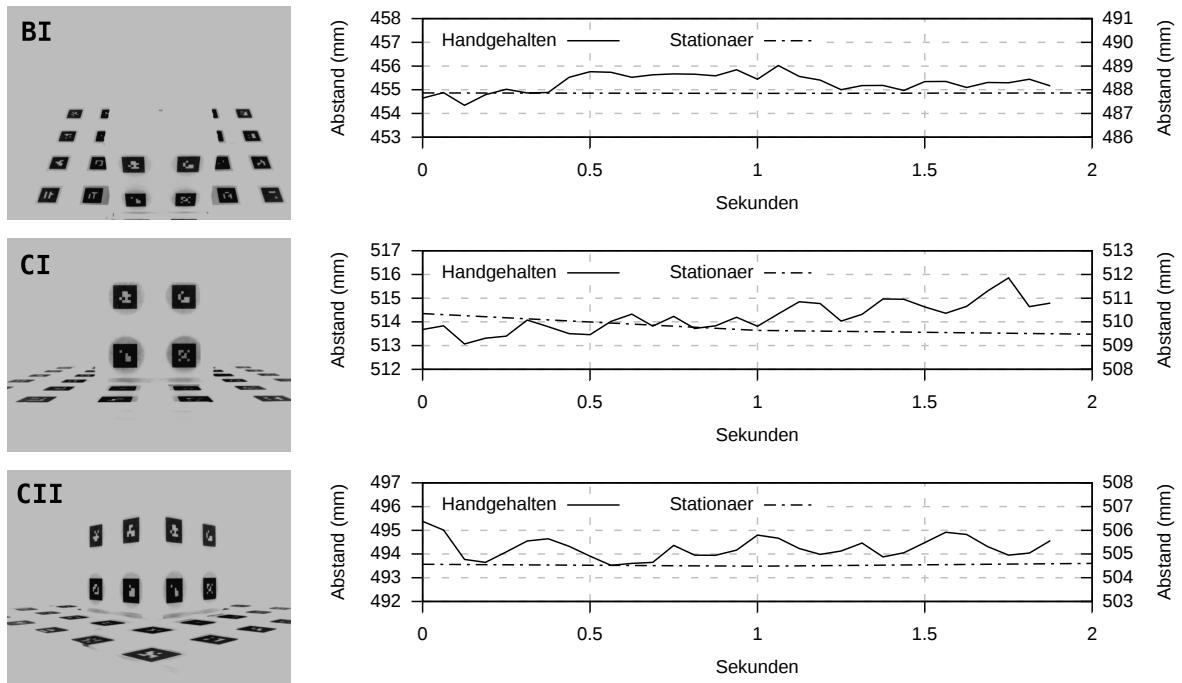


Abbildung 5.7.: Positions berechnung im Dunkeln: Die Diagramme zeigen die berechnete Distanz zwischen der Laptopkamera und dem Weltursprung in Millimeter. Es wurden dazu eine Videosequenz aus drei unterschiedlichen Blickwinkeln aufgenommen, wobei der Laptop einmal von Hand gehalten, und einmal stationär betrieben wurde. Die Bilder auf der linken Seite zeigen ein Frame des invertierten Webcamstreams. Es ist gut zu erkennen, dass im handgehaltenen Fall die Position um wenige Millimeter schwankt, was auf ein Verwackeln durch den Benutzers zurückzuführen ist. Im stationären Fall ist sie dagegen sehr stabil, das Sensorrauschen scheint keinen merklichen Einfluss zu haben. Man sieht, dass die Positions berechnung nicht immer exakt möglich ist: Im mittleren Graph driftet die Position im stationären Fall um rund einen Millimeter ab. Die Kamera wurde dabei nicht bewegt, weshalb dieser Effekt auf Ungenauigkeiten im Aufbau und Abweichungen in den Kameraparametern zurückzuführen ist.

Zusammenfassend lässt sich sagen, dass mit ARToolKit und der Tracking-Stage eine ausreichend genaue Positions berechnung möglich ist. Im weiteren Verlauf kann davon ausgegangen werden, dass die Position der Webcam bei der Beleuchtung zu jedem Zeitpunkt berechnet werden kann.

6. Beleuchtung

In diesem Kapitel wird erklärt, wie unter Zuhilfenahme des radiometrischen und geometrischen Modells aus Kapitel 4 eine hochauflösende Beleuchtung mit einem Bildschirm erzeugt werden kann. Im weiteren Verlauf wird davon ausgegangen, dass die Position und Orientierung des Bildschirms im Raum zu jedem Zeitpunkt bekannt ist. Die Berechnung erfolgt mit Hilfe der Tracking-Stage, ARToolKit und dem geometrischen Modell aus Abschnitt 4.1.

Damit mit dem Bildschirm an einer gegebenen Position eine einfallende Beleuchtung erzeugt werden kann, muss die Strahldichte berechnet werden, die von jedem Pixel in Richtung des Ursprungs emittiert werden soll. Diese gesamte, in die Szene einfallende Strahldichte wird dabei von einer Environment-Map vorgeben, anhand der sich, mit Hilfe des geometrischen Modells, die *benötigte* Strahldichte jedes einzelnen Bildschirmpixels berechnen lässt.

Daraufhin kann man das radiometrische Modell anwenden und für jeden Bildschirmpixel über das inverse Ansprechverhalten $v_{x,y}(l)$ den benötigten Framebuffer-Werte berechnen, sodass die gewünschte einfallende Strahldichte erzeugt wird. Befindet sie sich innerhalb des vom Bildschirm darstellbaren Bereichs, so kann man damit direkt eine Beleuchtung erzeugt. Der Dynamikbereich dieser Beleuchtung entspricht dabei genau dem des Bildschirms, weshalb sie hier als "LDR-Beleuchtung" bezeichnet wird.

Das Kontrastverhältnis einer echten Umgebungsbeleuchtung ist jedoch weitaus größer als das eines gewöhnlichen Bildschirms. In diesem Kapitel wird deshalb eine Methode vorgestellt, mit der sich der Dynamikbereich der erzeugten Beleuchtung durch eine zeitvariante Bildsequenz erhöhen lässt. Dieses Verfahren wird im Weiteren als "HDR-Beleuchtung" und die Sequenz als "HDR-Sequenz" bezeichnet.

Damit eine Szene vollständig, also aus allen gewünschten Richtungen beleuchtet werden kann, müssen mit dem Bildschirm viele einzelne Teilbeleuchtungen erzeugt werden. Dabei ist es wichtig, dass in jeder Teilbeleuchtung nur Licht emittiert wird, das noch nicht von einer anderen, vorangegangenen Teilbeleuchtung erzeugt wurde. Es muss also zu jedem Zeitpunkt bekannt sein, welcher Teil der Beleuchtung schon hergestellt wurde und welcher noch nicht.

Desweiteren soll die Beleuchtung von Hand durch einen Benutzer stattfinden, der den Bildschirm in einem dunklen Raum an die richtigen Stellen bewegt, und an jeder Position für mehrere Sekunden verharrt. Der Bildschirm muss dabei möglichst präzise, mit einem festen Abstand und Winkel, zur Szene hin ausgerichtet werden. Aus diesem Grund wird ein in diesem Kapitel auch ein Beleuchtungsablauf vorgestellt, bei dem der Benutzer von einem Computerprogramm mit Audiosignalen geführt wird, die ihm eine falsche Bildschirmausrichtung signalisieren und ihn so bei der Beleuchtung unterstützen. Erst wenn sich der Bildschirm an eine akzeptablen Position befindet wird die Teilbeleuchtung hergestellt und aufgenommen.

6.1. Benötigte Strahldichte

Sei L_r nun die Matrix der *benötigten Strahldichte*. Sie gibt an, welche Strahldichte jedes Bildschirmpixel im Ursprung O erzeugen soll. Der Abstand r_l zwischen Ursprung und Bildschirm, sowie der Ausrichtungswinkel sind dabei, genau wie bei der radiometrischen Kalibrierung in Abschnitt 4.2, fix.

Da die Positionen der Pixel im Raum bekannt sind, kann man L_r prinzipiell anhand einer beliebigen Environment-Map berechnen: Die Weltkoordinaten der als Punktlichtquellen modellierten Pixel geben dabei die Richtung an, mit der das emittierte Licht in den Ursprung einfällt. Mit dieser Lichtrichtung kann man dann die Strahldichte der Lichtquellen in der Environment-Map nachschlagen.

Sind die Punktlichtquellen regelmäßig auf einer Ebene angeordnet wie es bei einem Bildschirm der Fall ist, so ist es Sinnvoll eine Cube-Map als Environment-Map zu verwenden: Da Cube-Maps ebenfalls aus Ebenen bestehen, kann man L_r mit perspektivischen Projektionen berechnen. Insbesondere ist es auch möglich, den Projektionsprozess umzukehren, sodass man L_r zurück auf eine Cube-Map projizieren kann. Damit lässt sich berechnen, welcher Teil der Environment-Map von einer Teilbeleuchtung hergestellt wurde.

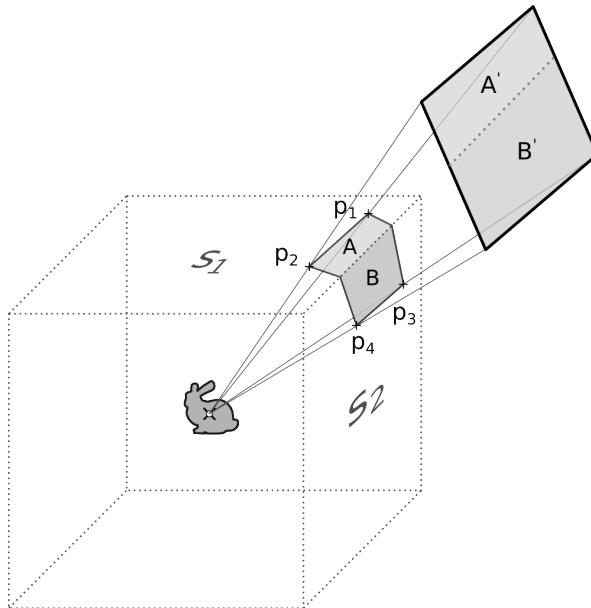


Abbildung 6.1.: Die benötigte Strahldichte L_r der Pixel wird mit Zentralprojektionen aus einer Cube-Map berechnet. Hierzu wird jede der sechs Seiten auf die Bildschirmebene projiziert, wobei der Augpunkt der Projektion im Ursprung O liegt. Die einzelnen Abbildungen ergeben aufsummiert die Strahldichteverteilung L_r . In der abgebildeten Bildschirmposition sind zwei Projektionen notwendig: Der Bereich A von Seite S_1 wird auf A' , und B von Seite S_2 wird auf B' abgebildet. Das Ergebnis ist $L_r = A' + B'$.

Zur Berechnung von L_r wird jede der sechs Cube-Map-Seiten einzeln auf den Bildschirm projiziert und das Ergebnis aufaddiert (Abbildung 6.1) Der Augpunkt der Projektion befindet sich dabei im Ursprung O . Die perspektivische Projektion ist vollständig definiert, wenn für mindestens vier Punkte auf der Cube-Map-Ebene bekannt ist, auf welche Punkte der Bildschirmebene sie abgebildet werden. Wie in Abbildung 6.2 dargestellt werden hierzu die vier Ecken der Bildschirmebene verwendet. Die jeweils korrespondierenden Punkte auf einer Seite der Cube-Map können durch einen Schnitt der Geraden, die durch die Ecken und den Ursprung verlaufen, berechnet werden.

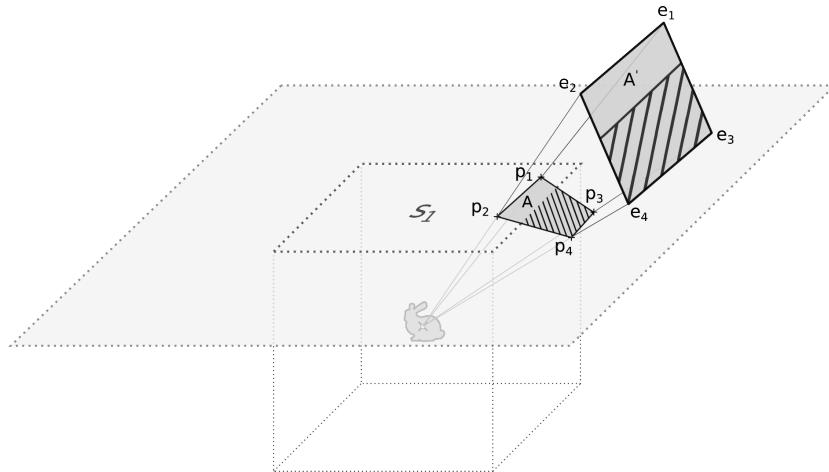


Abbildung 6.2.: Eine einzelne Seite S_1 der Cube-Map wird auf die Bildschirmebene projiziert: Die perspektivische Projektion ist dabei durch die vier Eckpunkte $e_1 - e_4$ des Bildschirms, und den vier Punkten $p_1 - p_4$ auf der Ebene s_1 festgelegt. Die Punkte p sind die Schnittpunkte der Geraden, die durch den Ursprung und die jeweilige Bildschirmecke e verlaufen. Sie liegen dabei zum Teil (p_3, p_4) auch außerhalb der eigentlichen Cube-Map-Textur (graue Ebene). In diesen Bereichen dürfen nur 0-Werte abgebildet werden (schraffierte Flächen).

Mit diesem Schema kann man also L_r anhand einer Environment-Map berechnen. Der Prozess lässt sich aber auch umkehren: Eine gegebene Bildschirmebene kann auf eine Cube-Map abgebildet werden. Hierzu wird einfach die Rolle der zwei Ebenen in der Projektion vertauscht und die Bildschirmebene einmal auf jede Cube-Map-Seite abgebildet. Damit ist es möglich, die von einem Bildschirm erzeugte Strahldichte in Form einer Environment-Map darzustellen und zu speichern. Dies wird für das korrekte Behandeln von Überlappungen beim Erzeugen einer vollständigen Beleuchtung in Abschnitt 6.4 benötigt.

6.2. LDR-Beleuchtung

Hat man die Strahldichteverteilung L_r berechnet, so kann man sie mit einem radiometrisch kalibrierten Bildschirm erzeugen - vorrausgesetzt die Werte befindet sich innerhalb des darstellbaren Bereichs. Dieser Bereich wird durch die minimale Strahldichte L_0 und die maximale Strahldichte L_{255} begrenzt,

6. Beleuchtung

die die Pixel im Ursprung erzeugen können, wenn man den Framebuffer auf $V = 0$ und $V = 255$ setzt. Damit eine LDR-Beleuchtung erzeugt werden kann, muss also

$$L_0 \leq L_r \leq L_{255} \quad (6.1)$$

gelten, wobei mit " \leq " der elementweise Vergleich gemeint ist: Die Ungleichung muss für jedes Element in der Matrix erfüllt sein. Ist dies der Fall, so kann man den Framebuffer V berechnen, indem man das inverse Ansprechverhalten auf jedes Pixel einzeln anwendet:

$$\forall(x, y) : V(x, y) = v_{x,y}(L_r(x, y)) \quad (6.2)$$

In Abschnitt 4.2.2 wurde das inverse Ansprechverhalten $v_{x,y}$ nicht pro Pixel, sondern pro Patch berechnet und liegt somit nur in einer niedrigen räumlichen Auflösung vor. Die Ansprechkurven jedes Pixels werden deshalb mit Hilfe einer Interpolation berechnet. Das kann zur Laufzeit geschehen, sodass nicht alle Kurven im Speicher gehalten werden müssen.

Zur Erklärung des Interpolation wird nun Abbildung 6.3 betrachtet. Sei $v_{(p_{i,j})}(l)$ das inverse Ansprechverhalten eines Patches $p_{i,j}$, und $(x_{i,j}, y_{i,j})$ sein Mittelpunkt in Pixelkoordinaten. Für die Interpolation wird angenommen, dass $v_{x_{i,j}, y_{i,j}}(l) = v_{(p_{i,j})}(l)$ gilt. Dem mittleren Patchpixel wird also genau das Ansprechverhalten zugeschrieben, das durch Mittelung der benachbarten Pixel berechnet wurde.

Die Ansprechkurve eines Pixels an der Position (x, y) kann dann durch eine bilineare Interpolation aus den vier Kurven der benachbarten Patches berechnet werden. Die vier Patchnachbarn eines Pixels sind die Patches, dessen Mittelpunkt am nächsten an (x, y) liegt.

Seien $p_{n,m}, p_{n+1,m}, p_{n,m-1}, p_{n+1,m-1}$ nun die vier Patchnachbarn des Pixels (x, y) . Die Kurve $v_{x,y}(l)$ des Pixels berechnet sich dann durch eine bilineare Interpolation von $v_{p_{n,m}}(l), v_{p_{n+1,m}}(l), v_{p_{n,m-1}}(l), v_{p_{n+1,m-1}}(l)$. Die Gewichtungen in der Interpolation sind dabei durch den Abstand des Pixels zu den vier Patchmittelpunkten gegeben.

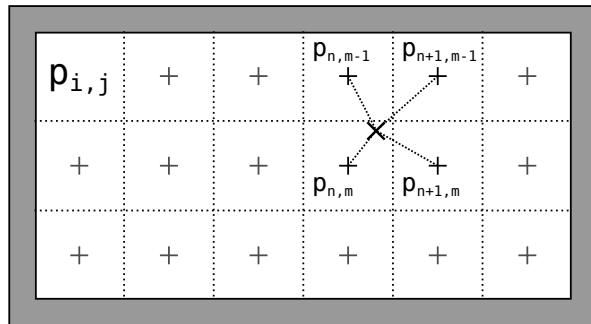


Abbildung 6.3.: Bilineare Interpolation des Per-Pixel-Ansprechverhaltens aus dem Per-Patch-Ansprechverhalten: Es wird angenommen, dass das Ansprechverhalten eines Patches $p_{i,j}$ für den jeweiligen Mittelpunkt (+) gültig ist. Die Ansprechkurve eines beliebigen Pixels (×) kann dann anhand einer bilinearen Interpolation aus den Kurven der benachbarten Patches berechnet werden.

Der Dynamikbereich einer LDR-Beleuchtung orientiert sich an der minimalen und der maximalen Strahldichte die mit dem Bildschirm herstellbar ist. Hier ist es wichtig, dass nur die Strahldichte

betrachtet wird, die von *jedem* Pixel erzeugt werden kann. Der Dynamikbereich R , den man mit einer LDR-Beleuchtung erreichen kann, lautet also

$$R = \frac{\min(L_{255})}{\max(L_0)} \quad (6.3)$$

wobei $\min()$ und $\max()$ das kleinste und das größte Element der Matrizen bezeichnet.

Im nächsten Abschnitt wird gezeigt, dass es durch Aufnahme eines Darkframes möglich ist, $L_0 = 0$ zu erreichen. Dies ist eine Voraussetzung für die HDR-Beleuchtung im darauf folgenden Abschnitt.

6.2.1. Reduktion von L_0

Die minimale Strahldichte L_0 , die ein Bildschirm erzeugen kann, entspricht seinem Schwarzwert und begrenzt den Dynamikbereich maßgeblich. Da das Licht des Bildschirms jedoch nicht von einem Menschen, sondern von einer Kamera wahrgenommen werden soll, kann L_0 bei fixer Bildschirmposition durch ein zweites Foto vermessen, und von der eigentlichen Aufnahme abgezogen werden. Das Prinzip ist analog zu der in Kapitel 4.2.2 vorgestellten Darkframe-Aufnahme, jedoch dass in diesem Fall die Hintergrundbeleuchtung angeschaltet bleibt und der Framebuffer auf $V = 0$ gesetzt wird.

Im Rest der Arbeit wird davon ausgegangen, dass zu jeder fotografierten Teilbeleuchtung T auch ein Darkframe D aufgenommen wird. Subtrahiert man das Darkframe von der eigentlichen Aufnahme T , so erhält man das Bild T' :

$$T' = T - D \quad (6.4)$$

Mit der Darkframesubtraktion ändert sich die effektiv herstellbare, minimale und maximale Strahldichte des Bildschirms:

$$\begin{aligned} L'_0 &= L_0 - L_0 = 0 \\ L'_{255} &= L_{255} - L_0 \end{aligned}$$

Der Wertebereich des Ansprechverhaltens $l_{x,y}(v)$ eines Pixels (x, y) verschiebt sich ebenfalls um $l^0_{x,y} = L_0(x, y)$. Das inverse Ansprechverhalten lautet also

$$v'_{x,y}(l) = v_{x,y}(l + l^0_{x,y}) \quad (6.5)$$

und kann zur Laufzeit aus $v_{x,y}(l)$ berechnet werden.

Der Dynamikbereich des Bildschirms verändert sich bei einer Darkframesubtraktion ebenfalls, da zum Einen der kleinste darstellbare Wert nicht mehr vom Schwarzwert $V = 0$, sondern von $V = 1$ bestimmt wird (siehe Definition des Dynamikbereichs in Abschnitt 2.3), und zum Anderen auch die maximale Strahldichte verringert wird. Sei L_1 nun die Strahldichte, die bei einem Framebuffer $V = 1$ vom Bildschirm im Ursprung erzeugt wird. Die kleinste erzeugbare Strahldichte $L'_1 > 0$ ist dann durch

$$L'_1 = L_1 - L_0 \quad (6.6)$$

6. Beleuchtung

gegeben. Der Dynamikbereich eines Bildschirms, im Falle einer Darkframesubtraktion, lautet also

$$R' = \frac{\min(L_{255} - L_0)}{\max(L_1 - L'_0)} = \frac{\min(L'_{255})}{\max(L'_1)} \quad (6.7)$$

Ob $R' > R$ oder $R' < R$ gilt hängt von dem Verhältnis zwischen L_1 und L_0 ab. Bei LCD-Bildschirmen ist der Schwarzwert L_0 meist deutlich größer als L'_1 , sodass $R' > R$ ist.

Es sei anzumerken dass diese Betrachtungsweise nur dann korrekt ist, wenn das Sensorrauschen der Kamera vernachlässigbar klein ist. Dies ist in der Praxis mit einem großen Aufwand verbunden, da es sich bei L_0 und insbesondere bei L'_1 um sehr kleine Lichtmengen handelt. Im weiteren Verlauf wird deshalb davon ausgegangen, dass alle Aufnahmen rauschfrei sind.

6.3. HDR-Beleuchtung

Der Dynamikbereich einer LDR-Beleuchtung ist stark eingeschränkt, besonders wenn ein Bildschirm mit starker Blickwinkelabhängigkeit verwendet wird. Es ist jedoch möglich das Kontrastverhältnis mit einer zeitvarianten Sequenz zu vergrößern: Während einer Langzeitbelichtung von mehreren Sekunden lassen sich nämlich nicht nur ein einzelnes, sondern gleich viele unterschiedliche Bilder nacheinander anzeigen. Bildlich gesprochen kann man so die Strahldichte L_r "über die Zeit verteilen". Es ist also eine feinere Regelung der emittierten Strahldichte möglich - insbesondere lassen sich so Werte kleiner als L'_1 herstellen.

Im Folgenden wird gezeigt, wie man so eine zeitvariante Sequenz konstruieren kann und wie sich damit der Dynamikbereich der erzeugten Beleuchtung vergrößern lässt. Sie wird hier als "HDR-Sequenz" bezeichnet, ein einzelnes Bild innerhalb der Sequenz als "Frame".

Angenommen wir haben einen Bildschirm und setzen den Framebuffer auf $V = 1$. Unter Einsatz der Darkframesubtraktion aus Abschnitt 6.2.1 messen wir, bei einer Belichtungszeit von t Sekunden, die Strahldichteverteilung L'_1 im Ursprung. Zeigt man in der Zeit t statt einem statischen Bild eine zeitvariante Sequenz aus m Frames an, die dabei gleichmäßig über die Zeit verteilt sind, so ist die kleinste Strahldichte, die in *einem Frame* hergestellt werden kann, L'_1/m , und die größte ist L'_{255}/m . Dies ist möglich, da mit der Darkframesubtraktion aus Kapitel 6.2.1, in einem Frame auch *gar kein* Licht emittiert werden kann ($L'_0 = 0$).

Würde man also L'_1/m mit einer HDR-Sequenz der Größe m erzeugen, so wäre genau in einem der Frames $V = 1$, und in allen anderen $V = 0$. Bei L'_{255} hingegen wäre in allen Frames $V = 255$, die maximale erzeugbare Strahldichte ändert sich also nicht.

Der Dynamikbereich, der mit einer HDR-Sequenz der Länge m erreicht werden kann, lautet also

$$R_{HDR} = \frac{\min(L'_{255})}{\max(L'_1)/m} = mR' \quad (6.8)$$

und ist damit genau m mal größer als R' .

Im Folgenden wird ein Algorithmus vorgestellt, mit dem eine HDR-Sequenz berechnet werden kann. Dazu wird die Belichtungszeit t als gegeben betrachtet. Sie wird vom Benutzer festgelegt, und orientiert sich an der Zeit die für eine einzelne Teilbeleuchtung zur Verfügung steht. Für die Anzeige der HDR-Sequenz kann die höchstmögliche Framerate $1/t_{max}$ gewählt werden, die mit dem Anzeigegerät ruckelfrei darstellbar ist. Es wichtig, dass jedes Frame genau t/m Sekunden auf dem Bildschirm verweilt, da sonst entweder zu viel oder zu wenig Licht emittiert wird.

Die Länge der HDR-Sequenz, die sich in der Zeit t erzeugen lässt, ist also durch

$$m = t/t_{max} \quad (6.9)$$

gegeben und somit auch festgelegt.

Die Aufgabe des Algorithmus ist es, eine gegebene Strahldichte L_r auf eine HDR-Sequenz der Länge m zu verteilen. Die Grundidee des Verfahrens soll anhand eines einfachen, eindimensionalen Beispiels verdeutlicht werden. In den Graphen in Abbildung 6.4) ist die Strahldichte von sechs fiktiven Pixeln dargestellt. Auf der linken Seite befindet sich die zu erzeugende Strahldichte L , die vom HDR-Algorithmus auf die drei Frames auf der rechten Seite verteilt wird.

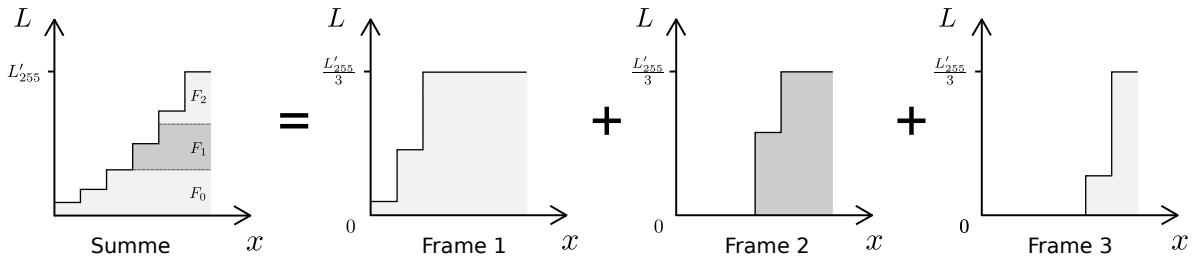


Abbildung 6.4.: Der HDR-Algorithmus verteilt die Strahldichte L , hier für sechs Pixel dargestellt, auf drei Frames. Jedes Frame wird genau $t/3$ Sekunden lang angezeigt, sodass in einer Langzeitbelichtung von t Sekunden in der Summe genau L erzeugt wird.

Damit L_r auch innerhalb der gegebenen Zeit t erzeugt werden kann, muss sie skaliert werden. Hierzu wird ein Faktor f so gewählt dass

$$\frac{L_r}{f} \leq L'_{255} \quad (6.10)$$

gilt. Der Dynamikbereich von L_r wird quasi in den der HDR-Sequenz eingepasst, sodass eine Beleuchtung mit dem höchstmöglichen Kontrast erzeugt werden kann, die mit dem Bildschirm in m Frames möglich ist. Den Faktor f berechnet man, indem das größte Element der elementweisen dividierten Matrizen L_r/L'_{max} sucht:

$$f = \max\left(\frac{L_r}{L'_{255}}\right) \quad (6.11)$$

Der HDR-Algorithmus basiert auf einem einfachen iterativen Verfahren, bei dem L_r/f schrittweise auf die Frames verteilt wird. Dazu wird eine temporäre Matrix $L_t = L_r/f$ initialisiert, und in einer Schleife m -mal $L_t = L_t - L'_{255}/m$ berechnet. In jedem Schritt lässt sich dann ein Frame aufbauen, indem man über alle Pixel (x, y) iteriert und die "restliche", noch nicht emittierte Strahldichte $L_t(x, y)$ betrachtet.

6. Beleuchtung

Dabei müssen drei unterschiedliche Fälle behandelt werden:

- Falls $L_t(x, y) \geq L'_{255}/m$ ist, so muss $V(x, y) = 255$ sein.
- Falls $L_t(x, y) \leq 0$ ist, so muss $V(x, y) = 0$ sein.
- In allen anderen Fällen ist $L_t(x, y)$ vom Bildschirm darstellbar: $V(x, y) = v'_{x,y}(L_t(x, y))$

Betrachtet man den Wert $V(x, y)$ eines Pixels über alle Frames einer so erzeugten HDR-Sequenz, so ist er zuerst über mehrere Frames $V(x, y) = 255$, dann wird in genau einem Frame einmal das inverse Ansprechverhalten $v'_{x,y}$ angewendet, und in den restlichen Frames ist $V(x, y) = 0$. Der Algorithmus lässt sich also optimieren indem man die zwei Schleifen vertauscht, und in der äußeren über die Pixel statt über die Frames iteriert. Das hat den Vorteil, dass abgebrochen werden kann sobald ein Pixel die zu erzeugende Strahldichte erreicht hat.

Der HDR-Algorithmus kann also wie im folgenden Pseudocode dargestellt implementiert werden:

```

 $\forall n < m : F_n \leftarrow 0$                                 // Alle Frames auf Null setzen
 $L \leftarrow \frac{1}{f} L_r$                                     // Strahldichte skalieren
for  $(x, y) = (0, 0)$  to  $(w - 1, h - 1)$  do                // für alle Pixel ..
    for  $n = 0$  to  $(m - 1)$  do                            // für alle Frames ..
        if  $L(x, y) \geq \frac{1}{m} L'_{255}(x, y)$  then          // maximale Strahldichte emittieren
             $F_n(x, y) \leftarrow 255$ 
             $L(x, y) \leftarrow L(x, y) - \frac{1}{m} L'_{255}(x, y)$ 
        else if  $L(x, y) < \frac{1}{m} L'_{255}(x, y)$  then      // Ansprechverhalten anwenden
             $F_n(x, y) \leftarrow v'_{x,y}(L(x, y))$ 
             $L(x, y) \leftarrow 0$ 
        else                                              // kein Licht emittieren
             $F_n(x, y) \leftarrow 0$ 
            break                                         // Abbruch, sobald  $L(x, y) = 0$ 
        end if
    end for
end for

```

Die Ausgabe des HDR-Algorithmus, die für den Laptopbildschirm berechnet wurde, ist in den Abbildungen 6.5 und 6.6 zu sehen. In 6.5 kann man sehen, dass mit einer HDR-Sequenz automatisch die ungleichmäßige Hintegrundbeleuchtung des Bildschirms ausgeglichen wird. Abbildung 6.6 zeigt hingegen eine HDR-Sequenz der Länge 50, wobei L_r von einer echten HDR-Umgebungsbeleuchtung vorgegeben wurde. Der dargestellte Ausschnitt besitzt dabei einen Dynamikbereich von 10000:1.

Der Algorithmus wurde für diese Arbeit mit OpenCV auf der CPU implementiert. Er kann aber auch problemlos auf einer GPU, beispielsweise als Pixel-Shader, implementiert werden. Eine CPU-basierte Implementierung ist für den Beleuchtungsablauf, der im nächsten Abschnitt 6.4.2 vorgestellt wird, jedoch vollkommen ausreichend, da ausreichen Rechenzeit zur Verfügung steht

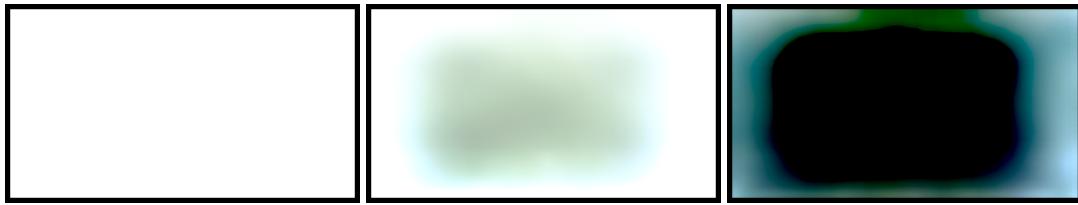


Abbildung 6.5.: Ausgabe vom HDR-Algorithmus: Eine uniforme Strahldichte ($L_r = 1$) wird auf drei Frames verteilt. Man kann gut erkennen, dass in den Randbereichen mehr Licht erzeugt wird als in der Mitte, da der Bildschirm dort eine niedrigere maximale Strahldichte L_{255} besitzt (siehe Abbildung 4.9).

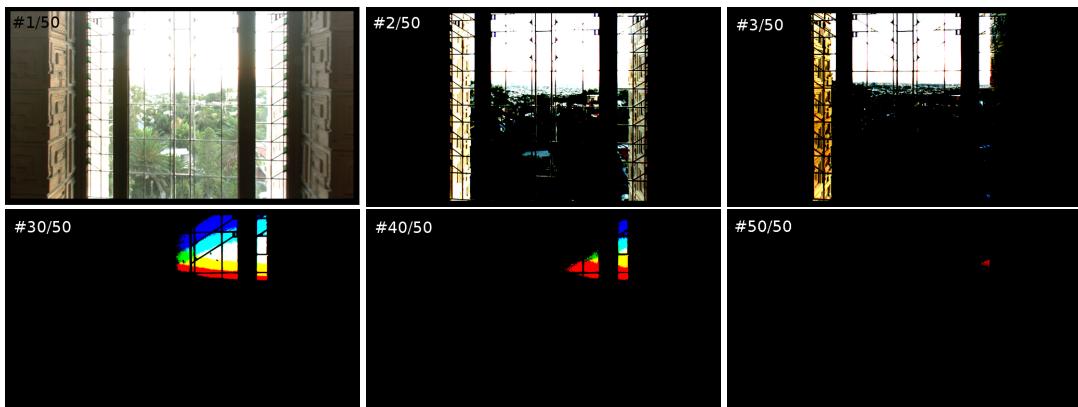


Abbildung 6.6.: Ausgabe vom HDR-Algorithmus: Dargestellt sind die Pixelwerte (V) von sechs Frames (1, 2, 3, 30, 40 und 50) einer HDR-Sequenz der Länge 50. Bei der Environment-Map handelt es sich um das “Ennis-Brown House” aus derDebevec HDR Light Probe Gallery [04]. Der Ausschnitt zeigt ein Raum, durch dessen Fenster die Sonne hineinscheint. Der Wertebereich umfasst dabei 5 Größenordnungen. Es ist gut zu sehen dass die Pixel, die eine größere Strahldichte erzeugen sollen, über einen längeren Zeitraum Licht emittieren. Kleine Strahldichtewerte werden dagegen mit den ersten Frames der Sequenz hergestellt. Es kommt dabei zu einer Auf trennung der Farben, da die Farbkanäle unabhängig behandelt werden.

6.4. Vollständige Beleuchtung

Ist die Bildschirmposition im Raum bekannt, so kann mit den in Abschnitt 6.1 - 6.3 vorgestellten Methoden eine einzelne Teilbeleuchtung erzeugt werden. Wenn sich der Bildschirm dabei auf der vorgesehenen Halbkugel befindet, so kann seine Position zu jedem Zeitpunkt anhand der Tracking-Stage aus Kapitel 5 berechnet werden. Die Szene kann somit durch einen Benutzer aus jeder Richtung beleuchtet werden.

Bevor mit diesem System jedoch eine vollständige, zusammenhängende Beleuchtung erzeugt hergestellt kann sind noch mehrere Hindernisse zu bewältigen: Zum Einen darf in jeder Teilbeleuchtung nur das Licht emittiert werden, das noch nicht von einer vorangegangenen Bildschirmposition erzeugt

6. Beleuchtung

wurde. Es muss also protokolliert werden, welche Teile der Beleuchtung schon hergestellt wurden und welche noch nicht. Ansonsten kann es passieren, dass ein Teil mehrfach erzeugt und somit zu viel Licht emittiert wird - die Teilbeleuchtungen *überlappen* sich.

Zum Anderen muss der Benutzer den Bildschirm immer mit dem Abstand und dem Winkel zur Szene hin ausrichten, die bei der radiometrischen Kalibrierung festgelegt wurden. Da der Beleuchtungsprozess "blind" in einem abgedunkelten Raum stattfindet, wird der Benutzer durch ein Programm geführt.

Im nächsten Abschnitt wird nun zuerst das Behandeln von Überlappungen erklärt, und danach auf den Beleuchtungsablauf eingegangen.

6.4.1. Behandeln von Überlappungen

In jeder Teilbeleuchtung darf immer nur der Teil der Beleuchtung hergestellt werden, der noch nicht in einer Vorangegangenen Bildschirmposition erzeugt wurde. Da die Tracking-Stage und die Positionsberechnung gewissen Toleranzen unterliegt, kann die Bildschirmposition in der Praxis nie exakt berechnet werden und weicht immer von der tatsächlichen Position ab. Man kann also nie genau wissen, welcher Teil der Beleuchtung tatsächlich erzeugt werden muss beziehungsweise erzeugt wurde. Des Weiteren ist der Bildschirm nicht fix - der Benutzer wackelt und driftet während der Beleuchtung langsam von der initialen Position ab. In der Praxis kommt es also, selbst wenn Buch über die Beleuchtung geführt wird, zu Überlappungen bei Teilbeleuchtungen.

Überlappungen äußern sich dadurch, dass an einer Stelle zu viel Licht und an einer anderen Stelle zu wenig Licht abgegeben wird. Wenn sich zwei Beleuchtungen überlappen, so wird im schlimmsten Fall an einer Stelle die doppelte Strahldichte erzeugt, und an einer anderen gar keine. Auf spiegelnden Oberflächen und im Backdrop werden sie deshalb als helle und dunkle Artefakte sichtbar.

Ein sehr ähnliches Problem tritt bei der Kalibrierung von Multi-Projektor-Systemen auf: Dort sollen mehrere Projektoren so angeordnet und kalibriert werden, dass sie gemeinsam ein zusammenhängendes Bild erzeugen. Es wird üblicherweise durch einen weichen Übergang (engl: "feathering") zwischen den Projektobildern gelöst [RBY⁺99]. Die Pixel, die sich überlappen werden hierzu mit einer linearen Rampe oder einer Cosinusfunktion multipliziert, die so gewählt ist, dass bei der Überlappung in der Summe der geforderte Helligkeitswert erzeugt wird. In der Praxis ist es meist ausreichend, wenn linear zwischen den Projektobildern übergegangen wird [RBY⁺99].

In dieser Arbeit werden einfache lineare Rampe verwendet. Da es bei einer vollständigen Beleuchtung an jeder Kante der Anzeigefläche zu einer Überlappung kommt, werden alle vier Ränder mit linearen ausgebendet. Die Breite der Rampen sollte dabei deutlich größer sein als die Abweichung, die bei der Positionsberechnung auftreten.

Sie werden werden in einer Alphamaske M gespeichert und bei Beginn der Beleuchtung vorberechnet. Bei der Strahldichteberechnung wird L_r dann elementweise mit M multipliziert, und die Matrix

$$L'_r = L_r \cdot M \quad (6.12)$$

berechnet, die als die Eingabe für den HDR-Algorithmus dient.

Dieser Vorgang ist in Abbildung 6.7 als Flussdiagramm dargestellt. Dabei bezeichnet E die Environment-Map, welche die Beleuchtung vorgibt, und P steht für die Zentralprojektionen aus Kapitel 6.1. Die untere Hälfte des Diagramms zeigt wie berechnet wird, welcher Teil der Umgebungsbeleuchtung erzeugt wurde: Die Maske M wird zuerst invertiert und dann mit der inversen Projektion P^{-1} (siehe Abschnitt 6.2) von der Bildschirmebene auf eine Cube-Map E_M abgebildet. E_M gibt an, welcher Anteil der Environment-Map in einer Teilbeleuchtung erzeugt wurde. Die Beleuchtung, die noch erzeugt werden muss, ergibt sich dann durch eine elementweise Multiplikation von E und E_M . Sie dient als Eingabe für die nächste Teilbeleuchtung.

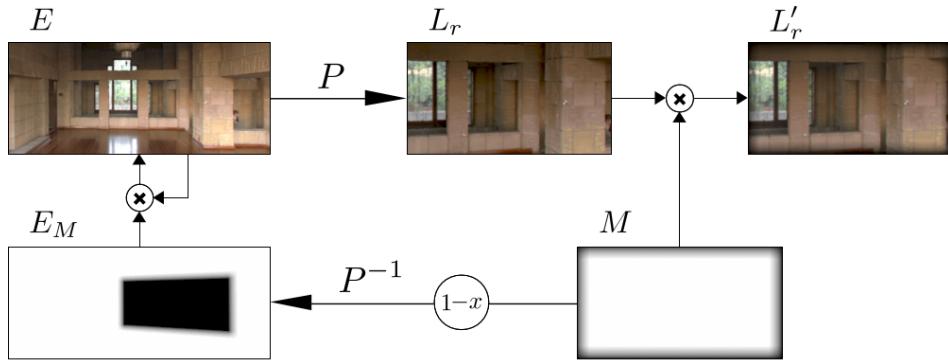


Abbildung 6.7.: Die Überlappungsbehandlung als Flussdiagramm: Ausgehend von einer Cube-Map E wird mittels perspektivischen Projektionen (hier mit P bezeichnet) die Strahlidichteverteilung L_r berechnet. Multipliziert mit der Maske M ergibt sich die Matrix L'_r , die als Eingabe für den HDR-Algorithmus dient. Die invertierten Masken ($1 - M$) wird mit einer inversen Projektion P^{-1} auf eine Cube-Map E_M abgebildet. Sie gibt an, welche Teile von E in den folgenden Teilbeleuchtungen erzeugt werden müssen. Mit $E = E \cdot E_M$ ist also die Beleuchtung für den nächsten Durchlauf gegeben.

6.4.2. Ablauf

Im Folgenden wird der Ablauf des Beleuchtungsprozesses erläutert. Er ist dabei in Form eines einzigen Programmes implementiert, das auf dem Laptop läuft. Es steuert alle Hardwarekomponenten des Systems (Bildschirm, Webcam, DSLR-Kamera) und berechnet, erzeugt und fotografiert die Teilbeleuchtungen. Es sorgt nebenbei auch dafür, dass die Annahmen die in der radiometrischen Kalibrierung getroffen wurden, zu jedem Zeitpunkt mit einer gewissen Toleranz eingehalten werden.

Abbildung 6.8 zeigt den Ablauf einer vollständigen Beleuchtung als Flussdiagramm. Jede Teilbeleuchtung wird dabei auf die gleiche Art und Weise durchgeführt: Als erstes bewegt der Benutzer den Bildschirm an eine Position und hält ihn ruhig. Die Position muss dabei die Annahmen erfüllen, die bei der radiometrischen Kalibrierung getroffen wurden. Der Abstand zwischen Szenenursprung und Bildschirmmittelpunkt muss r_l betragen, und die Ebene muss mit dem richtigen Winkel ausgerichtet sein. Ist dies nicht der Fall, so wird ein Audiosignal ausgegeben das den Benutzer anweist, wie er eine akzeptable Position erreichen kann (“näher kommen”, “weiter entfernen” und “Winkel einstellen”). Sobald die Position sich innerhalb einer gewissen Toleranz befindet, beginnt die Aufnahme einer

6. Beleuchtung

Teilbeleuchtung. Die DSLR wird dabei direkt vom Programm über USB gesteuert, wozu ein separater Thread eingesetzt wird.

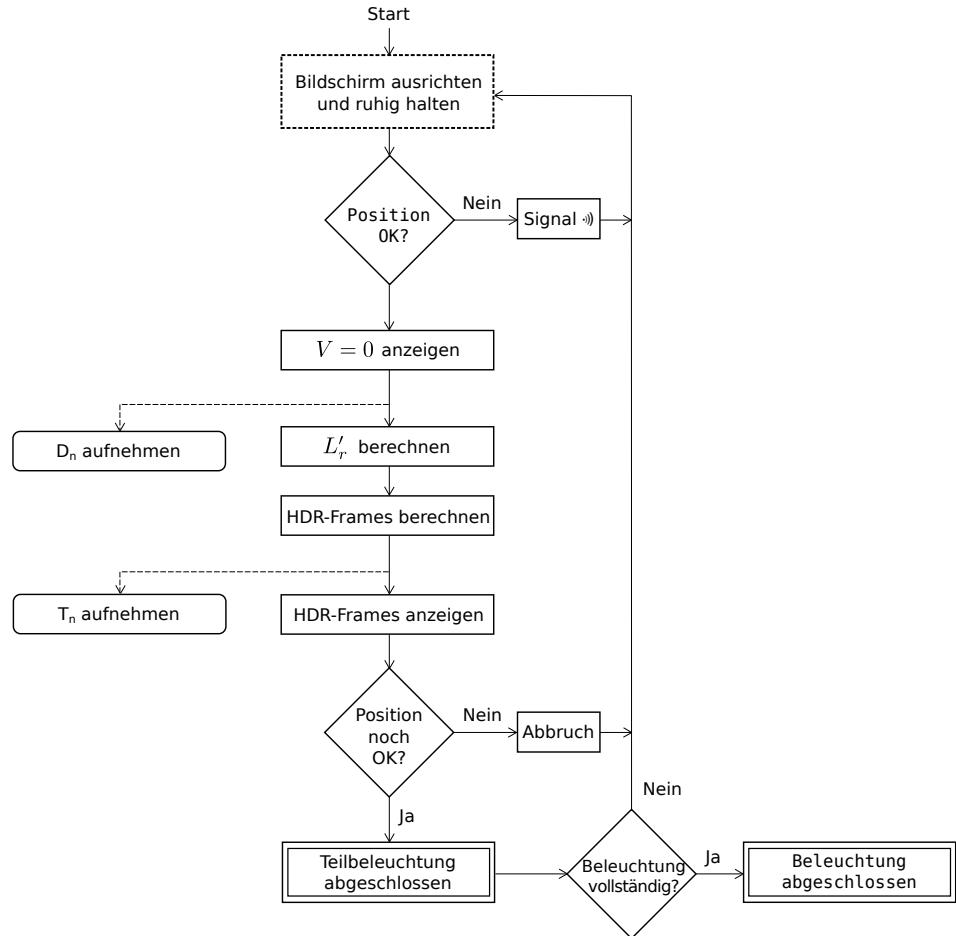


Abbildung 6.8.: Beleuchtungsablauf als Flussdiagramm: In jedem Durchlauf wird eine Teilbeleuchtung erzeugt. Dazu bewegt der Benutzer den Bildschirm an eine Position und hält ihn dort ruhig. Weicht die Position zu stark von der Vorgabe (Abstand, Winkel) ab, so wird ihm signalisiert wie er die richtige Position finden kann. Sobald er sie erreicht hat, wird der Bildschirm auf Schwarz geschaltet und L'_r sowie die HDR-Frames berechnet. Gleichzeitig wird, in einem separaten Thread, das Darkframe aufgenommen. Im Anschluss wird die HDR-Sequenz angezeigt und aufgenommen. Falls der Benutzer dabei abdriftet oder verwackelt, so wird die Teilbeleuchtung abgebrochen und muss erneut durchgeführt werden. Andernfalls wird die erzeugte Beleuchtung von der Environment-Map abgezogen (siehe Kapitel 6.4.1), und der Prozess beginnt von neuem. Der Vorgang ist abgeschlossen, sobald alle gewünschten Teilbeleuchtungen erzeugt sind.

6.4. Vollständige Beleuchtung

Die Berechnung von L'_r (Kapitel 6.1) sowie der HDR-Algorithmus (Kapitel 6.3) läuft parallel zur Darkframeaufnahme ab. In dieser Zeit muss der Laptop lediglich ein schwarzes Bild ($V = 0$) anzeigen, weshalb die Rechenzeit für die Berechnung der HDR-Frames verwendet werden kann.

Sobald alle Frames im Speicher vorliegen und die Darkframeaufnahme abgeschlossen ist, kann die HDR-Sequenz angezeigt und die Teilbeleuchtung Fotografiert werden. In dieser Zeit hat das Programm wenig zu tun - es muss lediglich dafür sorgen, dass die Frames mit einer konstanten Framerate angezeigt werden. Im Flussdiagramm ist, aufgrund der einfacheren Darstellung, eingezeichnet, dass die Position am Ende der Teilbeleuchtung noch einmal kontrolliert wird. In der tatsächlichen Implementierung geschieht dies jedoch kontinuierlich, zu jedem Zeitpunkt, in einem weiteren separaten Thread. Sobald der Benutzer abdriftet oder zu stark verwackelt, wird die Teilbeleuchtung abgebrochen und beide DSLR-Aufnahmen verworfen.

Befindet sich die Bildschirmposition während der gesamten Teilbeleuchtung innerhalb einer vorgegeben Toleranz, so gilt die Aufnahme als erfolgreich und die erzeugte Beleuchtung wird von der Environment-Map E abgezogen (Kapitel 6.4.1). Daraufhin beginnt der Prozess von neuem. Sobald alle gewünschten Teilbeleuchtungen erzeugt wurden kann der Beleuchtungsvorgang beendet, und das Endergebnis berechnet werden (siehe nächster Abschnitt).

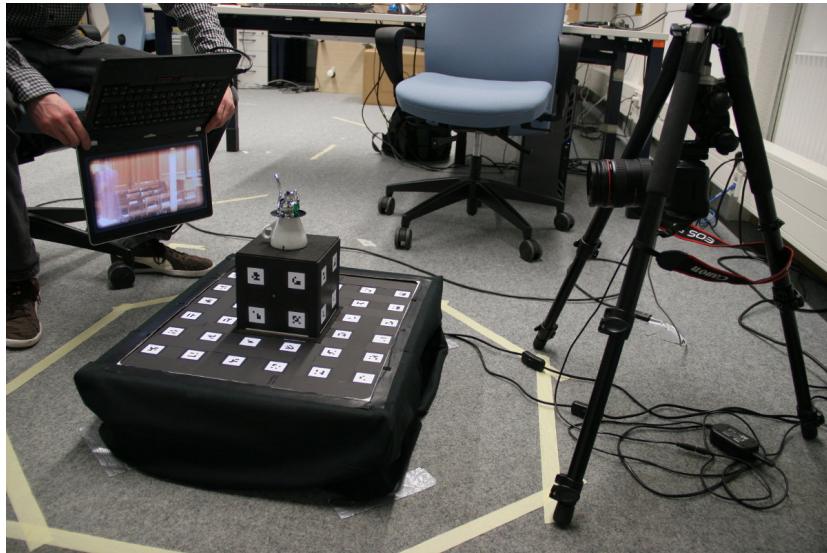


Abbildung 6.9.: Haltung des Benutzers bei der Beleuchtung: Die Tracking-Stage wird auf dem Boden aufgebaut, wodurch eine Sitzende Position möglich ist. Der Benutzer kann die Arme dabei auf den Knien aufstützen, sodass der Bildschirm sehr ruhig gehalten werden kann.

Um Verwackler und ein Driften zu reduzieren, kann die Tracking-Stage auf dem Boden aufgebaut werden. Der Benutzer kann so auf einem Stuhl sitzen, und die Arme auf den Knien aufstützen (Abbildung 6.9), wodurch der Bildschirm deutlich ruhiger gehalten werden kann als es beispielsweise mit gestreckten Armen der Fall ist. Damit kann ein Großteil der Beleuchtung ziemlich genau und ohne Mühe erzeugt werden. Die allerobersten Positionen können jedoch nur im Stehen mit ausgestreckten Armen erreicht werden.

6. Beleuchtung

6.4.3. Zusammenfügen der Aufnahmen

Nachdem alle Teilbeleuchtungen erzeugt und fotografiert sind, müssen die Aufnahmen zu einem Endergebnis zusammengefügt werden. Sei T_i die aufgenommene Szene unter der i -ten Teilbeleuchtung, D_i das zugehörige Darkframe, und T_S das Endergebnis.

Im Folgenden wird angenommen, dass alle T_i und D_i mit einer radiometrisch Kalibrierten DSLR-Kamera (siehe Abschnitt 4.2.1) aufgenommen, und mithilfe dem invertierten Ansprechverhalten in relative Strahldichte umgerechnet wurden. Wenn dies der Fall ist, so können die Aufnahmen durch eine Addition zu einem Endergebnis kombiniert werden.

Hier kommt nun der Faktor f ins Spiel, mit dem die Strahldichte beim Berechnen der HDR-Sequenz skaliert wurde: Mit ihm kann diese Skalierung wieder “rückgängig” gemacht werden, sodass die Strahldichte der Aufnahmen T_i im korrekten Verhältnis zueinander stehen.

Sei f_i der Skalierungsfaktor f , der in der i -ten Teilbeleuchtung berechnet wurde. Das Endergebnis T_S , das die Szene unter dem Einfluss aller Teilbeleuchtungen zeigt, kann durch Auswertung der gewichteten Summe

$$T_S = \sum_i f_i (T_i - D_i) \quad (6.13)$$

berechnet werden.

Im Rest des Kapitels werden nun die einzelnen Teilespekte des vorgestellten Verfahrens evaluiert.

6.5. Evaluation

Die einzelnen Aspekte des Verfahrens werden getrennt evaluiert:

LDR-Beleuchtung

Evaluation der radiometrischen Kalibrierung. Es wird gezeigt, dass sich mit dem Bildschirm eine vorgegebene Strahldichte L_r im Ursprung erzeugen lässt. Dabei wird auch die Winkelabhängigkeit untersucht, da der handgeholtene Bildschirm in der Praxis nicht exakt ausgerichtet werden kann.

HDR-Beleuchtung

Evaluation der HDR-Sequenz. Es wird untersucht, ob der Dynamikbereich der Beleuchtung mit der zeitvarianten Sequenzen auch tatsächlich vergrößert werden kann.

Überlappungen

Es wird gezeigt, dass Überlappungen korrekt behandelt werden und sich eine zusammenhängende Lichtfläche erzeugen lässt.

Im Folgenden werden die mit der Kamera gemessenen einfallenden Strahldichteveverteilungen mit L_c bezeichnet. Zur Darstellung werden Heatmap-Diagramme verwendet, die den Luminanzkanal der Aufnahme zeigen (siehe Gleichung A.3). Da in den Diagrammen relative Strahldichte dargestellt wird, werden alle Messwerte einer Aufnahmereihe auf 1 normiert.

6.5.1. Evaluation: LDR-Beleuchtung

In diesem Abschnitt wird untersucht, ob sich mit dem radiometrisch kalibrierten Bildschirm eine LDR-Beleuchtung erzeugen lässt. Dazu wurde eine synthetische Strahldichteverteilung gewählt, die aus drei linearen Rampen besteht. Sie wurde mit einer HDR-Sequenz der Länge 1 erzeugt, sodass die Eingabe zuerst auf den Dynamikbereich des Bildschirms skaliert wird, und dann über das inverse Ansprechverhalten des Bildschirms abgebildet wird.

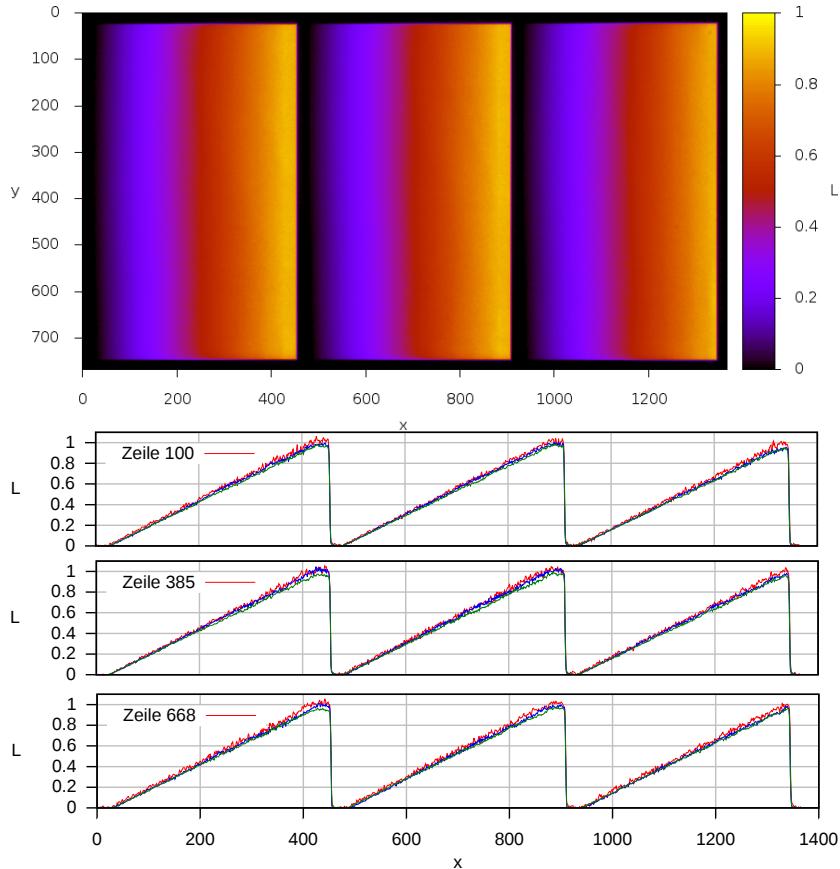


Abbildung 6.10.: Die Strahldichteverteilung L_c im Ursprung der Szene, beim Erzeugen von einer LDR-Beleuchtung. Als Vorgabe L_r wurden drei linearen Rampen verwendet, die jeweils mit einem Abstand voneinander getrennt sind. Die Linienplots zeigen die Strahldichte von drei Bildschirmzeilen. Die Messwerte zeigen, dass sowohl die radiometrische Kalibrierung als auch die Berechnung der LDR-Beleuchtung funktioniert: L_c entspricht weitgehend der linearen Eingabe. Es gibt jedoch eine geringe Abweichungen zwischen den Farbkanälen, die mit dem Kanalübersprechen (siehe Abschnitt 4.2) erklärt werden kann. Das Rauschen in den Messdaten wird weitgehend durch den Kamera-Sensor verursacht. Es kann jedoch nicht ausgeschlossen werden, dass auch der Bildschirm daran beteiligt ist.

6. Beleuchtung

Die so erzeugte Strahldichte wurde dann mit der DSLR-Kamera im Ursprung gemessen und ist in Abbildung 6.10 dargestellt. Damit winkelabhängige Effekte ausgeschlossen werden können, wurde die Aufnahme direkt im Anschluss zur radiometrischen Kalibrierung durchgeführt und die Kamera dazwischen nicht bewegt.

In der Praxis kann der handgehaltene Bildschirm nicht mit dem exakten Winkel ausgerichtet werden, weshalb die Annahmen die bei der Kalibrierung getroffen wurde, verletzt werden. Da der Blickwinkel einen Einfluss auf das Ansprechverhalten hat, wurde die LDR-Beleuchtung nicht nur im Ursprung, sondern auch unter verschiedenen Winkeln gemessen. Die Kamera wurde dabei jeweils 5° in die horizontale, und in die vertikale Richtung geneigt (siehe Abbildung 6.11).

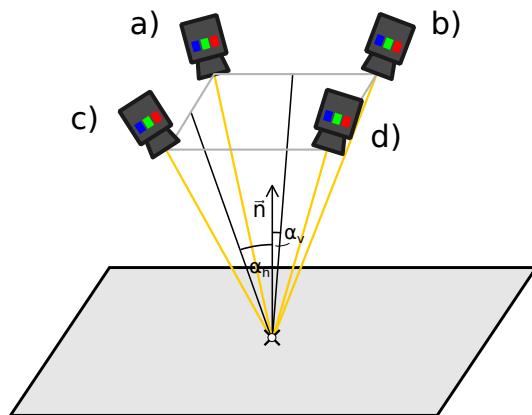


Abbildung 6.11.: Zur Evaluation der Blickwinkelabhängigkeit bei einer LDR-Beleuchtung wurde die Strahldichte L_c aus vier unterschiedlichen Positionen aufgenommen. Die Kamera wurde dazu bei der Aufnahme horizontal um $\alpha_h = \pm 5^\circ$ und vertikal um $\alpha_v = \pm 5^\circ$ geneigt.

Die vier Diagramme in Abbildung 6.12 zeigen L_c , gemessen aus den vier Positionen a) - d). Der Abstand betrug dabei $r_l = 50$ cm. Man sieht dass der Blickwinkel einen sehr großen Einfluss auf das Ansprechverhalten hat, da der Laptopbildschirm auf der TN-Technologie basiert. Nimmt man an, dass er bei der Beleuchtung immer so ausgerichtet ist, dass $|\alpha_h| < 5^\circ$ und $|\alpha_v| < 5^\circ$ gilt, dann ist mit einer Abweichung von $\pm 20\%$ zu rechnen.

6.5.2. Evaluation: HDR-Beleuchtung

In diesem Abschnitt soll gezeigt werden, dass die HDR-Sequenz auch in der Praxis funktioniert und mit ihr der Dynamikbereich des Bildschirms vergrößert werden kann. Hierzu wurde eine HDR-Beleuchtung mit verschiedenen Sequenzenlängen erzeugt und gemessen. Als Eingabe für den HDR-Algorithmus kam eine Exponentialrampe zum Einsatz, die Werte zwischen 10^{-3} und 1 beinhaltet, also einen Dynamikbereich von 1000:1 besitzt. Die erzeugte HDR-Beleuchtung wurde über die Reflexion an einer schwarzen Billardkugel gemessen, damit längere Belichtungszeiten und somit längere HDR-Sequenzen möglich sind. Die Kugel hatte dabei einen Durchmesser von 6 cm und wurde so platziert dass der Ursprung im Zentrum der Reflexion liegt.

6.5. Evaluation

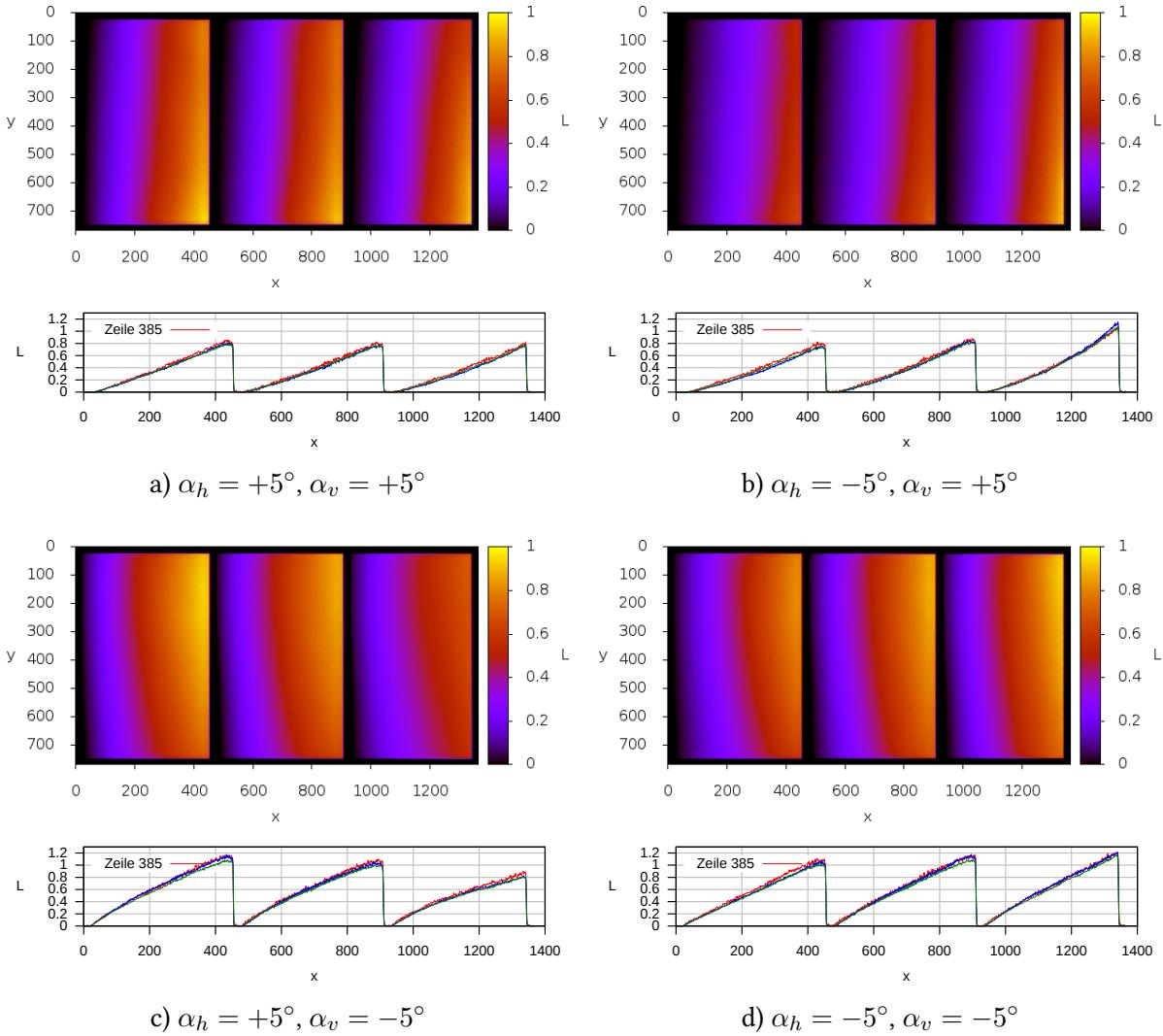


Abbildung 6.12.: Die LDR-Beleuchtung, gemessen aus den vier Blickwinkeln a)-d) (siehe Abbildung 6.11). Die Graphen zeigen jeweils die Strahldichte der mittleren Pixelzeile. Dass Ansprechverhalten ist stark winkelabhängig: Man kann deutlich erkennen, dass die vertikale Ausrichtung den *Verlauf* der Kurven beeinflusst. Die horizontale Ausrichtung wirkt sich dagegen auf die *maximale Strahldichte* aus.

Abbildung 6.13 zeigt die mit einer DSLR-Kamera gemessene Strahldichte für die Sequenzlängen 1, 10, 20, und 100. Da die Kamera einen eingeschränkten Dynamikbereich besitzt, wurde die längste Sequenz auch mit einer HDR-Aufnahme (10/20/30 Sekunden) aufgenommen, längere Belichtungszeiten waren in dieser Konfiguration nicht möglich.

6. Beleuchtung

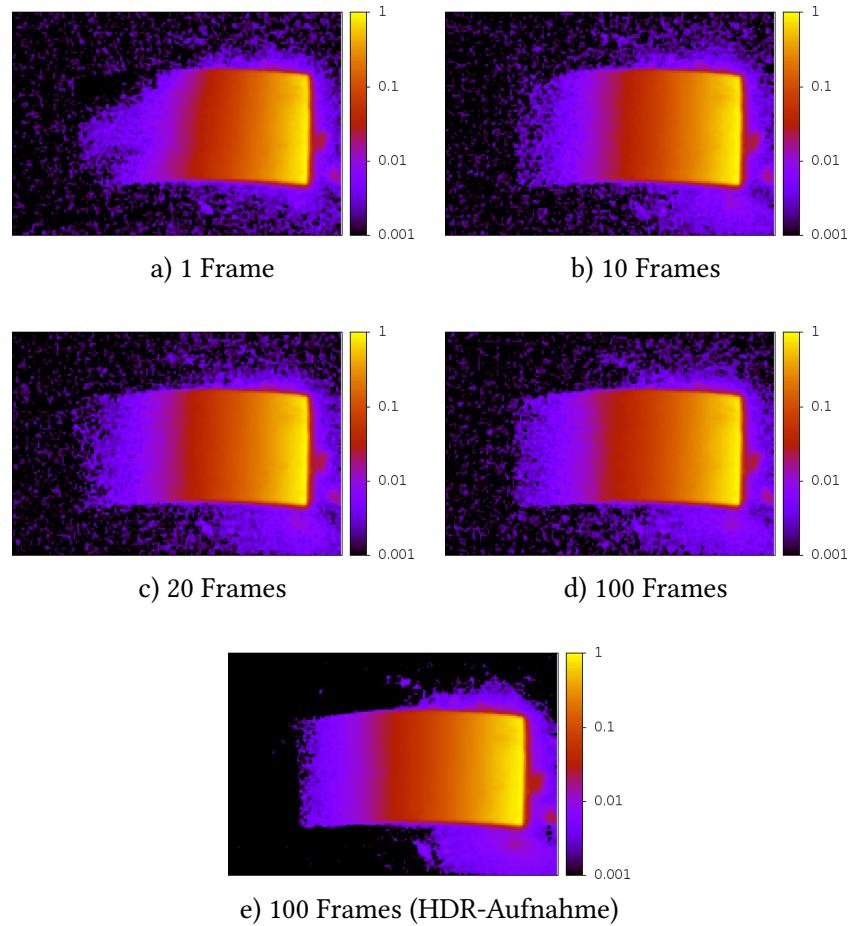


Abbildung 6.13.: Zur Evaluation wurden HDR-Beleuchtungen mit verschiedenen langen Sequenzen erzeugt und über die Reflexion an einer schwarzen Billardkugel aufgenommen. Als Eingabe L_r wurde eine Exponentialrampe verwendet, die Werten zwischen 10^{-3} und 1 beinhaltet ($R = 1000 : 1$). Die Heatmaps zeigen die Strahldichte auf einer logarithmischen Skala, weshalb die Rampe im Plot (im Optimalfall) einen linearen Verlauf besitzen muss. Man sieht, dass sich mit längeren Sequenzen kleinere Strahldichtewerte erzeugen lassen - der Dynamikbereich kann folglich vergrößert werden. Für eine echte HDR-Beleuchtung werden jedoch sehr viele Frames benötigt: Selbst mit $m = 100$ liegt der Dynamikbereich des Laptopbildschirms noch deutlich unter 1000:1.

Da die Framerate begrenzt ist, mit der die Sequenz angezeigt werden kann, wären für eine wirkliche HDR-Beleuchtung sehr lange Belichtungszeiten notwendig. Mit einem handgehaltenen Bildschirm ist dies in der Praxis, wenn der Beleuchtungsablauf aus Kapitel 6.4 verwendet wird, nicht umsetzbar, da der Benutzer häufig zu weit von der Position abdriften würde.

6.5.3. Evaluation: Überlappungen

Damit bei der vollständigen Beleuchtung eine zusammenhängende Lichtfläche ohne Lücken oder helle Flecken erzeugt werden kann, muss nicht nur die tatsächliche Bildschirmposition sehr genau bekannt sein, der Benutzer muss ihn theoretisch auch absolut still halten. Die Breite der linearen Rampen, mit den die Teilbeleuchtungen ineinander übergeblendet werden, hängt zum Einen von der Genauigkeit der Positionsbestimmung, und zum Anderen von dem Bewegungsspielraum ab, den man bei der Beleuchtung zulässt. Die Genauigkeit der Positionsbestimmung kann nur mit hohem Aufwand vollständig evaluiert werden, weshalb die Überlappungsberechnung nur anhand zweier Bildschirmpositionen untersucht wurde. Für eine vollständige Evaluation müsste nicht nur jede mögliche Bildschirmposition, sondern auch jeder Ausrichtungswinkel untersucht werden.

Zur Evaluation wurde eine Rampenbreite von 80 Pixel gewählt, was im Falle des Laptops 1,7 cm entspricht, und zwei überlappende Teilbeleuchtungen einmal mit Rampe, und einmal ohne erzeugt, und auf einem Chrom-Objekt fotografiert. Dabei wurde eine synthetische Environment-Map verwendet, sodass Lücken und Überlappungen deutlich sichtbar gemacht werden können.

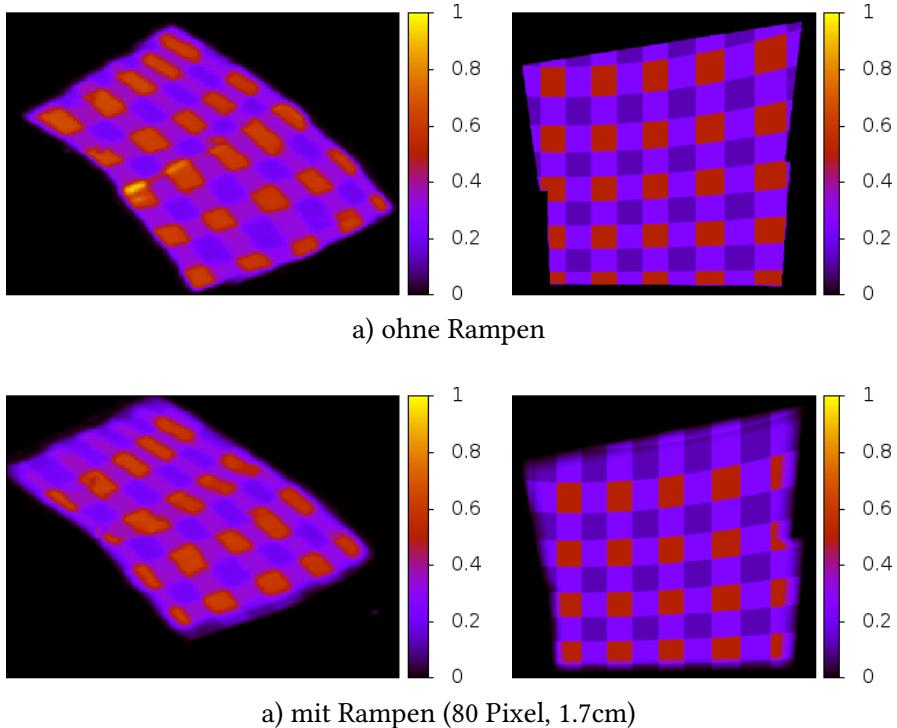


Abbildung 6.14.: Zwei überlappende Teilbeleuchtungen wurden anhand einer synthetischen Environment-Map erzeugt und über ein Chrom-Objekt, das sich im Ursprung befindet, fotografiert. Die linke Spalte zeigt die Aufnahmen, und die rechte den erzeugten Ausschnitt der Environment-Map. In **a)** wurden die Teilbeleuchtungen nicht ineinander überblendet, in **b)** wurde eine 80 Pixel breite Rampe verwendet, was in etwa 1.7cm auf dem Bildschirm entspricht. Der Bildschirm wurde dabei wie in Abbildung 6.9 dargestellt von Hand gehalten.

6. Beleuchtung

Der Bildschirm wurde dabei wie im Beleuchtungsablauf (siehe Abbildung 6.9) von Hand gehalten. In Abbildung 6.14 sind auf der linken Seite die Aufnahme, und auf der rechten Seite der Ausschnitt der Environment-Map dargestellt. Man kann sehen, dass ohne ein Überblenden zwischen den Teilbeleuchtungen deutlich sichtbare Artefakte entstehen können. Die Aufnahmen sind leider nicht direkt miteinander vergleichbar, da die Bildschirmposition nicht identisch sind. Es kann also keine Aussage über die optimale Breite s_r der Rampen getroffen werden.

7. Ergebnisse

In diesem Kapitel wird gezeigt wie sich das Beleuchtungsverfahren, dessen einzelne Teilespekte in Abschnitt 6.5 bereits getrennt evaluiert wurden, in der Praxis verhält. Hierzu wurden zwei Szenen mit den folgenden Eigenschaften ausgewählt:

- Szene 1: stark spiegelnde und transparenten Objekte ($r_s = 3\text{cm}$)
- Szene 2: gemischte Szene mit Schatten ($r_s = 5 \text{ cm}$)

Anhand Szene 1 (siehe Abschnitt 7.1) soll gezeigt werden, dass das Beleuchtungsverfahren funktioniert und es damit möglich ist, eine einfallende Beleuchtung mit einer hohen Auflösung zu erzeugen. Dazu wurde sie von allen Richtungen beleuchtet, die mit dem Laptop und der Tracking-Stage erreicht werden konnten. Damit die einzelnen Teilbeleuchtungen dabei möglichst schnell erzeugt werden können wurde eine kurze HDR-Sequenz gewählt und der Dynamikbereich der Environment-Map reduziert.

Mit Szene 2 (siehe Abschnitt 7.2) hingegen soll demonstriert werden, dass eine korrekte Beleuchtung nur dann möglich ist, wenn man lange HDR-Sequenzen verwendet. Eine vollständige HDR-Beleuchtung mit langen HDR-Sequenzen (150 Frames und mehr) würde jedoch viele Stunden dauern und ist damit nicht mehr praktisch anwendbar. Für Szene 2 wurde darum eine synthetische Environment-Map verwendet, und nur eine einzige Teilbeleuchtung erzeugt.

7.1. Ergebnis: Szene 1

Für Szene 1 wurde eine Environment-Map mit reduziertem Kontrast (etwa 200:1) verwendet, und eine HDR-Sequenz der Länge 20 gewählt. Damit konnte für die DSLR-Aufnahmen eine Belichtungszeit von 1.6 Sekunden gewählt werden, wodurch sich eine Teilbeleuchtung im Bestfall unter 10 Sekunden erzeugen ließ.

Die Toleranzen wurde so eingestellt, dass der Bildschirmabstand $r_l = 50 \text{ cm}$ vom Benutzer auf $\pm 5 \text{ cm}$ genau eingehalten werden muss. In so gut wie allen Teilbeleuchtungen konnte der Drift des Bildschirmzentrums geringer als 1 cm gehalten werden. Für die allerobersten Positionen, bei denen der Bildschirm mit ausgestreckten Armen gehalten werden muss, wäre es jedoch notwendig 2 cm Drift zu erlauben. Für den Winkel zwischen Bildschirmnormalen und dem Vektor \vec{f} wurden maximal 5° zugelassen; die allerunterste Reihe konnte jedoch nur bei einem Winkel von 7° erreicht werden. Insgesamt wurden 78 Teilbeleuchtungen erzeugt, wobei der ungenutzte Rand 23x24 Pixel groß war. Die Verwendete Anzeigefläche betrug damit 1320x720 Pixel.

7. Ergebnisse

Zur Beleuchtung wurde die in Abbildung 7.1 gezeigte Environment-Map gewählt. Die Bildschirmpositionen der einzelnen Teilbeleuchtungen sind ebenfalls in Abbildung 7.1 dargestellt.

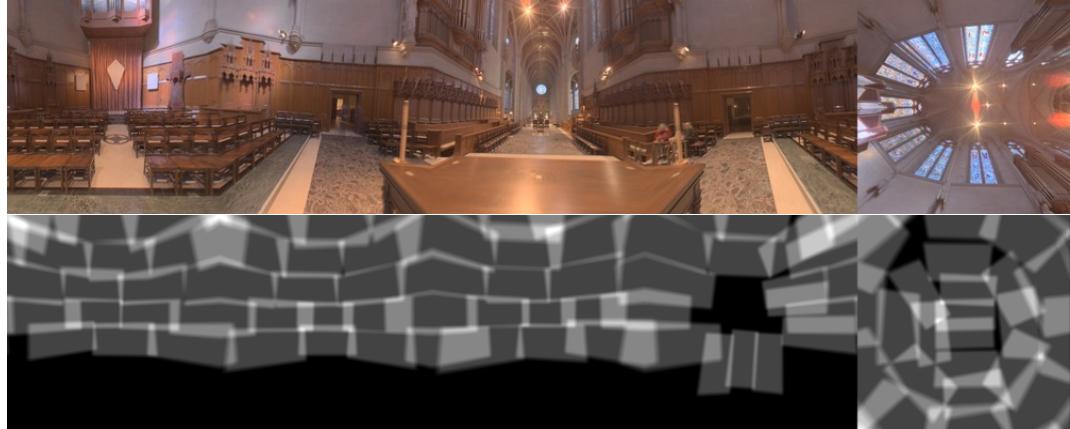


Abbildung 7.1: **Oben:** Die für Szene 1 verwendete “Grace”-Environment-Map entstammt der Debevec Light Probe Gallery [04]. Der Dynamikbereich wurde mit dem Adaptive-Logarithmic-Tonemapping Verfahren [DMAC03] auf etwa 200:1 reduziert. Die Auflösung jeder Textur beträgt 1000x1000 Pixel. **Unten:** Die Bildschirmpositionen und Überlappungen aller 78 Teilbeleuchtungen.

Abbildung 7.2 zeigt die Bildschirmzentren aller Teilbeleuchtung als 3D-Plot. Die Farbe gibt dabei den Index i der aufsteigend nummerierten Teilbeleuchtungen an. Man kann erkennen, dass der Backdrop als erstes hergestellt wurde (schwarz). Anschließend wurde der Bildschirm kreisförmig um die Szene, hin und her bewegt und die obere Hälfte der einfallen Beleuchtung von unten nach oben erzeugt. Die untersten Teilbeleuchtungen (gelb) sind als letztes entstanden.

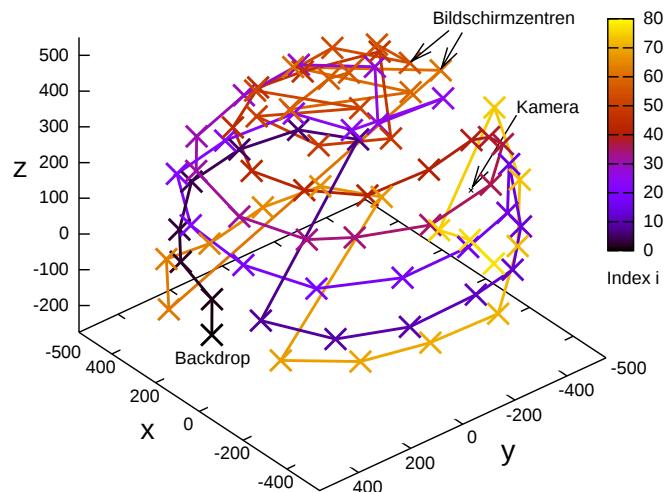


Abbildung 7.2: Die Position des Bildschirmzentrums (\vec{f}) bei jeder der 78 Teilbeleuchtungen von Szene 1 (in Millimeter). Die Farbe der Linie gibt die Reihenfolge an, in der die Beleuchtungen erzeugt wurden.

7.1. Ergebnis: Szene 1



Abbildung 7.3.: Szene 1 wurde mit der Environment-Map aus Abbildung 7.1 beleuchtet. Das Ergebnis wurde aus insgesamt 78 Teilbeleuchtungen zusammengesetzt. Die schwarzen Flächen sind Positionen, die mit dem Laptop nicht erreicht werden konnten (unteres Drittel der Environment-Map). Es ist gut zu erkennen, dass die meisten der Teilbeleuchtungen, trotz einer hohen Winkeltoleranz von 7° , eine optisch zusammenhängende Lichtfläche erzeugen. Überlappungen werden korrekt behandelt, wenn der Bildschirm ruhig gehalten wird. Im Bild sieht man jedoch auch mehrere Teilbeleuchtungen, die zu dunkel oder verrutscht sind. Diese Artefakte werden in Abbildung 7.4 genauer untersucht.

Dauer der Beleuchtung	
Anzahl Teilbeleuchtungen	78
Benötigte Zeit (gesamt)	95 Minuten
Benötigte Zeit (pro Teilbeleuchtung)	73 Sekunden
Durchschnittliche Laufzeiten	
Projektion P (Cube-Map auf Bildschirm)	64.32 ms
Projektion P^{-1} (Bildschirm auf Cube-Map)	77.37 ms
HDR-Algorithmus	796.08 ms
Positionsberechnung	4.82 ms

Tabelle 7.1.: Dauer der Beleuchtung von Szene 1 und durchschnittliche Laufzeiten.

7. Ergebnisse

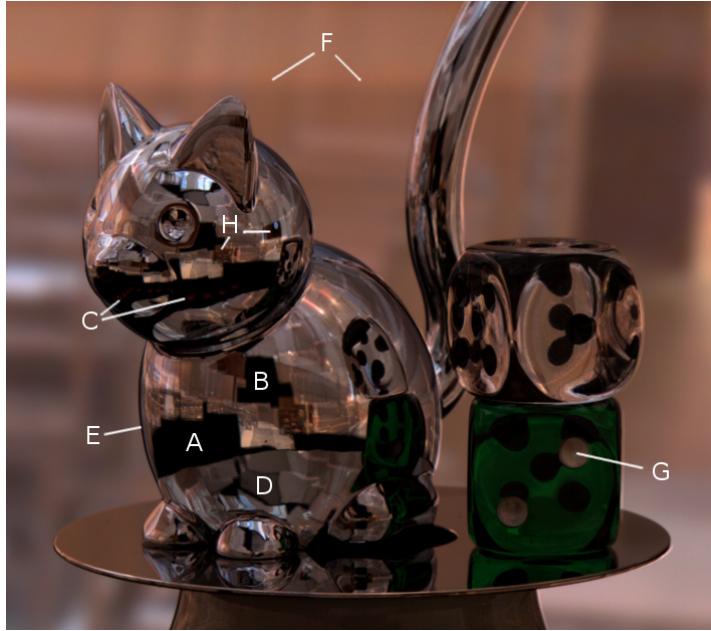


Abbildung 7.4.: Erklärung der Artefakte die im Ergebnisbild der Szene 1 zu sehen sind:

- **A:** Unerreichbare Position: Der Laptop kann nicht an diese Positionen bewegt werden, da die Fiducials der Tracking-Stage hier außerhalb des Webcamfrustums liegen.
- **B:** Position der DSLR-Kamera: Hier kann keine Beleuchtung erzeugt werden, da das Blickfeld der Kamera nicht verdeckt werden darf.
- **C:** Hier spiegeln sich die Fiducials: Sie sind als schwaches, rotes Rauschen sichtbar, weil sie vom Bildschirm beleuchtet werden.
- **D:** Oberste Bildschirmposition: Der Bildschirm musste hier mit ausgestreckten Armen gehalten werden, wodurch es zu einem stärkeren Drift kam.
- **E:** Die Kamera hat sich zwischen den Aufnahmen minimal verschoben: Der schwarze Rand sollte hier nicht sein.
- **F:** Überlappung im Backdrop: Verursacht durch eine ungenaue Position oder ein Drift (als heller Streifen in Abbildung 7.3 sichtbar).
- **G:** Diffuse Flächen sind verrauscht und zu dunkel: Der Dynamikbereich der DSLR-Kamera ist zu klein. Die Intensität des Lichts, das an diesen Stellen bei einer Teilbeleuchtung reflektiert wird, ist zu niedrig und kann deshalb nicht korrekt gemessen werden.
- **H:** Teilbeleuchtungen mit zu niedriger Intensität: Verursacht durch Probleme bei der Shuttersteuerung; der DSLR-Shutter löste zu spät aus.

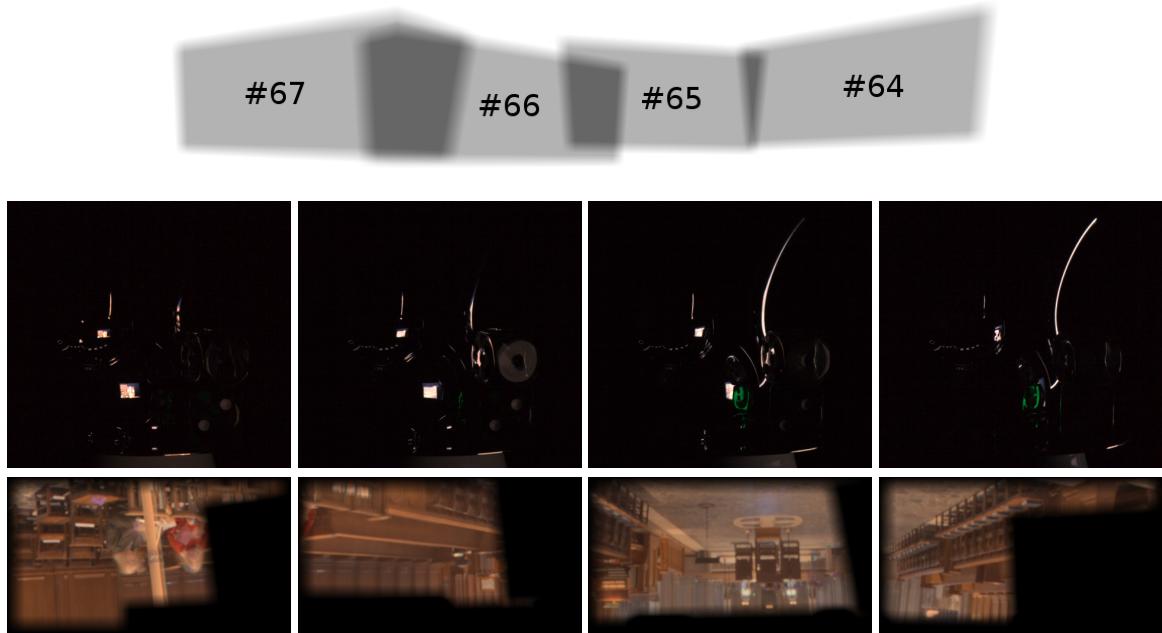


Abbildung 7.5.: Die 64.-67. Teilbeleuchtung. **Oben:** Die Bildschirmpositionen als Cube-Map; die Beleuchtungen wurden von rechts nach links erzeugt. **Mitte:** Aufnahme T_i (Dark-frame nicht abgezogen). Die leuchtenden Fiducials, sowie der hohe Schwarzwert des Bildschirms sind zu erkennen. **Unten:** Die StrahldichteVerteilung L_r der Teilbeleuchtungen. Durch die Überlappungen bleiben teilweise große Bereiche der Bildschirmfläche ungenutzt.

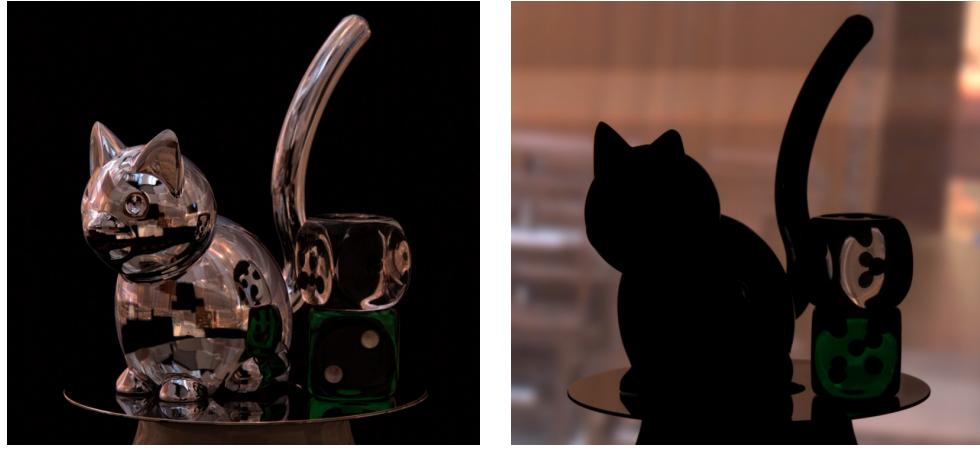


a) Ergebnis

b) reale Beleuchtung

Abbildung 7.6.: Vergleich zwischen dem Ergebnis (links) und einer echten Beleuchtung (rechts). Man sieht das die weißen, diffusen Würfelaugen unter Einfluss einer realen Beleuchtung deutlich heller sind als im Ergebnis (siehe Artefakt G in Abbildung 7.4).

7. Ergebnisse



a) ohne Hintergrund

a) nur Hintergrund

Abbildung 7.7.: Die Szene lässt sich ganz einfach vom Hintergrund trennen, indem nur bestimmte Teilbeleuchtungen zusammengeführt werden. Abbildung **a)** zeigt Szene 1 ohne Hintergrund, Abbildung **b)** hingegen zeigt nur den Hintergrund.

7.2. Ergebnis: Szene 2

Szene 2 besteht aus verschiedenen diffusen, glänzenden und transluzenten Objekten, die so angeordnet sind, dass Schatten und Kaustiken gut sichtbar gemacht werden können.

Für die Beleuchtung wurden zwei synthetische Environment-Maps gewählt. Beide bestehen aus einer einzigen kleinen, punktförmige Lichtquelle (weiß), die bei 50cm Entfernung etwa 2cm auf dem Bildschirm einnimmt. In der ersten Environment-Maps ist die Lichtquelle zusätzlich von einer großen Flächenlichtquelle (grün) mit niedriger Intensität umgeben. Die relative Strahldichte der Punktlichtquelle ist 500-mal größer als die der Flächenlichtquelle - das Kontrastverhältnis der Beleuchtung beträgt also 500:1. Die Länge der HDR-Sequenz wurde mit 160 Frames so groß gewählt, dass das dieses Kontrastverhältnis auch dargestellt werden kann.

Abbildung 7.8 zeigt Szene 2 unter beiden Beleuchtungen. Hierzu wurde jeweils einer HDR-Aufnahme aus drei Belichtungen (10/20/30 s) rekonstruiert. Es ist deutlich zu erkennen, dass auch Lichtquellen mit einer geringen Intensität einen signifikanten Einfluss auf das Erscheinungsbild haben können, wenn sie eine große Fläche einnehmen. Da eine echte Umgebungsbeleuchtungen sowohl große Flächen mit geringer Intensität, als auch kleine Flächen mit sehr hoher Intensität besitzt, kann man sie nur mit einer HDR-Beleuchtung wahrheitsgetreu erzeugen. Eine vollständige HDR-Beleuchtung ist mit dem in dieser Arbeit vorgestellten System aufgrund der langen HDR-Sequenzen leider nur unter großem Zeitaufwand möglich.

7.2. Ergebnis: Szene 2

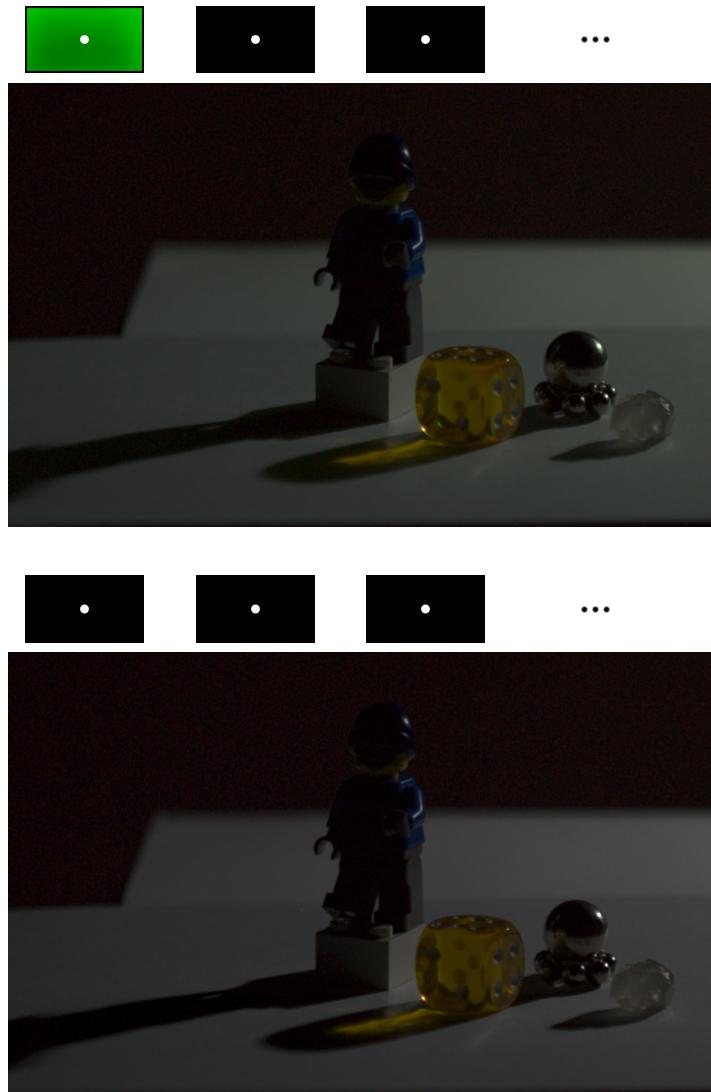


Abbildung 7.8.: Szene 2 wurde aus nur einer Bildschirmposition, mit zwei unterschiedlichen synthetischen Environment-Maps beleuchtet. Über den Aufnahmen sind die ersten Frames der HDR-Sequenz dargestellt. **Oben:** Punktlichquelle (2cm) vor einem grünen Flächenlichtquelle. Die weiße Lichtquelle ist 500-mal heller als die grüne ($R = 500 : 1$). **Unten:** Das selbe Szenario, jedoch ohne Flächenlichtquelle. Vergleicht man den Schatten in beiden Aufnahmen, so kann man erkennen, dass die grüne Lichtquelle, trotz ihrer geringen Intensität, noch einen signifikanten Einfluss auf das Erscheinungsbild hat. Eine HDR-Beleuchtung ist also wichtig für eine fotografische Beleuchtung: Es müssen gleichzeitig kleine Lichtquellen mit einer hohen Intensität, und große Lichtquellen mit einer geringen Intensität erzeugt werden.

8. Zusammenfassung und Ausblick

In dieser Arbeit konnte gezeigt werden dass es möglich ist, mit einem Bildschirm eine fotografische Beleuchtung in hoher Auflösung herzustellen.

Um dies zu realisieren, wurde das System aus Hardwarekomponenten zuerst geometrisch und radiometrisch modelliert und anschließend kalibriert. Dabei wurde für jedes Bildschirmpixel ein eigenes Ansprechverhalten angenommen, und im Kalibrierungsvorgang mit reduzierter Auflösung rekonstruiert. Es wurde eine Anordnung aus selbstleuchtenden Fiducials beschrieben, die es erlaubt, die Bildschirmposition über die im Laptop eingebaute Webcam zu bestimmen. Anhand der von ARToolKit berechneten Webcamposition konnte so über das geometrische Modell die Position aller Pixel in Relation zur Szene berechnet werden.

Die Strahldichte, die von jedem Pixel ausgehen muss, wurde mittels perspektivischen Projektionen aus einer Cube-Map berechnet. Durch eine invertierung des Ansprechverhaltens war es möglich eine LDR-Beleuchtung anhand einer Environment-Map zu erzeugen. Es wurde gezeigt, wie der Dynamikbereich des Bildschirms dabei mit einer zeitvariante Sequenz vergrößert werden kann, und wie sich eine solche Sequenz konstruieren und berechnen lässt.

Das vorgestellte Verfahren wurde stückweise evaluiert, und im Anschluss anhand zweier Testszenarios in der Praxis getestet. Mit den Ergebnissen konnte gezeigt werden dass es möglich ist, eine Szene von Hand in hoher auflösung nahezu vollständig zu beleuchten. Die Überlappungen von Teilbeleuchtungen wurde dabei korrekt behandelt, sodass eine zusammenhängende Lichtfläche entsteht. Da ein Benutzer bei der Beleuchtung driftet, verwackelt, und auch den Winkel nur ungenau einstellen kann, treten dennoch Artefakte auf, die eine weitere Optimierung des Systems bedingen.

Es konnte herausgefunden werden, dass mit dem Laptopbildschirm eine HDR-Beleuchtung nur mit sehr langen HDR-Sequenzen möglich ist. Aufgrund der beschränkten Framerate werden dazu sehr lange Belichtungszeiten benötigt, in der ein Benutzer den Bildschirm absolut still halten muss. Das Verfahren ist, so wie es beschrieben wurde, also nur bedingt in der Praxis einsetzbar.

Ausblick

Es sind zahlreiche Erweiterungen denkbar, mit denen sich die Ergebnisse und die Anwendung in der Praxis verbessern lassen könnte.

Tracking-Stage

Die Tracking-Stage ermöglicht zwar die Positions berechnung bei sehr vielen Bildschirmpositionen, eine vollständige Beleuchtung ist damit allerdings nicht möglich. Es sind andere Ansätze denkbar, beispielsweise könnte man den Bildschirm mit Hilfe von externen Kameras tracken

8. Zusammenfassung und Ausblick

oder eine zweite Kamera daran befestigen. Damit wäre der Aufbau zwar nicht mehr so kompakt, aber die Bildschirmposition wäre in jeder Position berechenbar, sodass auch der untere Teil der einfallenden Beleuchtung erzeugt werden kann.

Ansprechverhalten

Der Bildschirm kann in der Praxis nie exakt von Hand an die Position bewegt werden, für die er radiometrisch kalibriert wurde. Da der Winkel im radiometrischen Modell als fix betrachtet wird, kommt es zu Abweichungen in der hergestellten Beleuchtung. Die maximale Strahldichte sowie das Ansprechverhalten ändern sich.

Würde man das radiometrischen Modell um die Winkelabhängigkeit erweitern, so könnte man dieses Problem eliminieren. Der Benutzer hätte mehr Freiheit bei der Beleuchtung und enge Toleranzen müsste er nicht mehr einhalten: Die Beleuchtung wäre auch bei einem abweichendem Abstand oder Winkel korrekt.

Ein Modell, das den Austrittswinkel beachtet, ist um zwei Dimensionen größer als das beschriebene. Eine diskrete Darstellung der Ansprechkurven würde sehr viel Speicher benötigen, man müsste sie als parametrische Funktionen (Polynome oder Exponentialfunktionen) darstellen. Auch die Kalibrierungsroutinen wären deutlich komplizierter, da der Bildschirm aus vielen Blickwinkeln aufgenommen werden muss. Damit ein passendes Modell entwickelt werden kann, muss die Winkelabhängigkeit des Bildschirms genauer untersucht werden.

Beleuchtungsablauf

Das beschriebene Beleuchtungsverfahren hat den Nachteil, dass der mit einer HDR-Sequenz erreichbare Dynamikbereich in der Praxis aufgrund der beschränkten Framerate stark begrenzt ist. Es sind allerdings auch andere Vorgehensweisen denkbar, mit denen ein größerer Dynamikbereich möglich sein sollte. Man könnte beispielsweise eine HDR-Beleuchtung in Echtzeit durchführen, indem man die Strahldichteberechnung und den HDR-Algorithmus auf der GPU implementiert. Die DSLR-Kamera könnte man dabei im Bulb-Mode ansteuern. An den Stellen, an den die einfallende Beleuchtung eine hohe Intensität besitzt, muss der Bildschirm dann auch länger verweilen. Der Benutzer muss so den Bildschirm dann nicht mehr still an einzelne Positionen halten, sondern kann ihn langsam um die Szene bewegen. Ein Verwackeln oder Driften spielt bei einer ausreichend präzisen Positions berechnung keine Rolle mehr, da es in Echtzeit ausgeglichen werden kann.

In diesem Fall ist es jedoch nicht mehr möglich, den Benutzer mit einem Programm an die richtige Positionen zu führen. Eine Echtzeitbeleuchtung setzt deshalb ein um den Winkel erweitertes Ansprechverhalten, wie im vorherigen Abschnitt beschrieben wurde, voraus. Des Weiteren ist es so auch nicht möglich, ein Darkframe aufzunehmen. Um eine HDR-Beleuchtung in Echtzeit realisieren zu können müsste man also einen LED-Bildschirm einsetzen, da der hohe Schwarzwert des LCDs nicht mehr reduziert werden kann.

A. Anhang

A.1. Hardware

A.1.1. Laptop

Modellbezeichnung	Lenovo Thinkpad Edge E330
Prozessor	Intel(R) Core(TM) i3-3120M CPU @ 2.50 GHz
Arbeitsspeicher	3.5 GB
Festplatte	80GB SSD (Intel SSDSA2M080G2GC)
Gewicht	1,8 kg
Webcam	640 x 480 Pixel Fixfocus (intern) 16 FPS (im Dunkeln) FOV: 46.7° × 35.8°
Bildschirm	1366 x 768 Pixel 13"TN-Panel LED-Backlight 293.5 x 165 mm

Tabelle A.1.: Technische Daten des Laptops.

A.1.2. DSLR-Kamera

Modellbezeichnung	Canon EOS 5D Mark II
Sensor	21 MP (5616 x 3744) Full frame (36 x 24 mm) Typ: CMOS
Objektiv	Canon EF 24-70mm f/2.8L

Tabelle A.2.: Technische Daten der DSLR-Kamera

A. Anhang

A.1.3. Tracking-Stage Komponenten

Die Komponenten des Tracking-Stage Prototyps haben insgesamt, ohne die LED-Lampen, 35.98 EUR gekostet und sind in der folgenden Tabelle aufgeführt. Als Werkzeug wird lediglich ein Klebestift, eine 60mm Lochsäge und eine Bohrmaschine, sowie ein Laserdrucker zum Ausdrucken der Fiducials benötigt.

1x	Glasplatte	50x60 cm, 2mm stark
4x	Acrylglasplatte	15x15 cm, 2mm stark
1x	Holzkiste	15x15x15 cm
8x	Plastikbecher	13 hoch
1x	Blickdichter Stoff	100x120cm
2x	Philips hue [07] LED-Lampe	

Tabelle A.3.: Die Komponenten der Tracking-Stage.

A.2. Software

Der Großteil der Software, wie beispielsweise die Kalibrierungsroutinen und die Beleuchtungsberechnung, wurden in C++11 mit OpenCV [05] implementiert. Zur Steuerung des Framebuffers wurde OpenGL verwendet. Der Toolflow wurde mithilfe von Shell-Skripten implementiert. Alle Bibliotheken und Softwarepakete, die in dieser Arbeit verwendet wurden, sind in der folgenden Tabelle aufgeführt.

gcc 4.7.4 [13]	C++11 Compiler
OpenCV 2.4.7 [05]	Projektionen, Kamera-Kalibrierung, Matrixoperationen
ARToolKit 2.72.1 [08]	Marker-Tracking
gphoto2 2.5.3 [09]	Steuerung der DSLR-Kamera
dcraw 9.20 [11]	Umwandlung von RAW-Formaten, De-Bayering
ImageMagick convert [12]	Umwandlung und Bearbeiten von (HDR)Bildern
pfstools 1.8.5 [01]	Betrachten und bearbeiten von HDR-Daten
pfstmo 1.5 [02]	Tone-Mapping
pfscalibration 1.5 [03]	Rekonstruktion des Kamera-Ansprechverhaltens
OpenEXR [10]	Dateiformat für HDR-Daten

Tabelle A.4.: Verwendete Softwarepakete und Bibliotheken.

A.3. Ergänzende Abbildungen

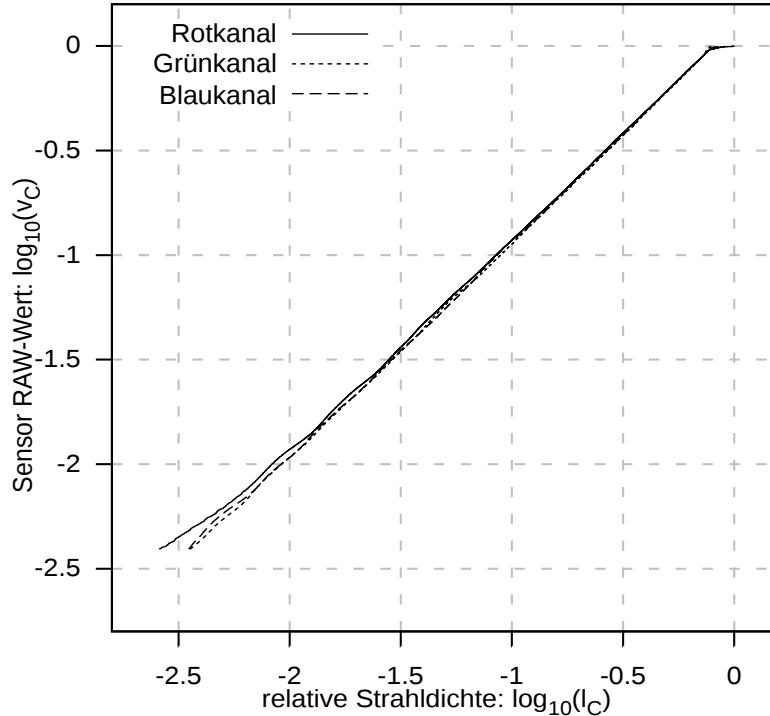


Abbildung A.1.: Doppellogarithmischer Plot des Ansprechverhalten einer Canon EOS 5D Mark II DSLR Kamera, rekonstruiert mit dem Programm pfscalibration [03]. Dabei ist die relative Strahldichte $\log_{10} l_C$ über die RAW-Werte $\log_{10} v_C$ abgetragen, wobei die Messwerte auf den Bereich (0, 1) normiert wurden.

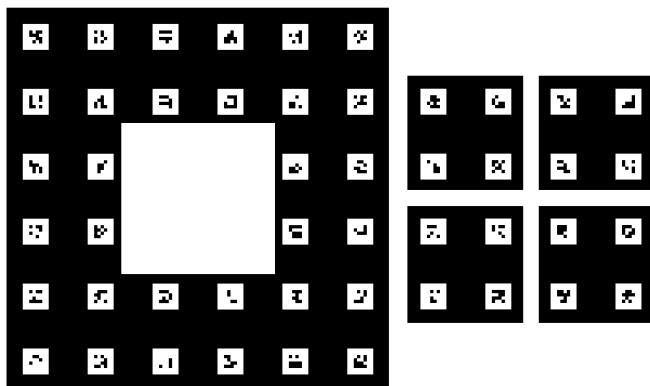


Abbildung A.2.: Die fünf Ebenen der Tracking-Stage: Die Fiducials sind 30x30 mm groß. Die horizontale Ebene (links) besitzt insgesamt 32 Marker und misst 45x45 cm. Die vier vertikalen Ebenen (rechts) sind 15x15 cm groß, und tragen jeweils vier Fiducials. Die Marker sind farblich invertiert, da sie sich so besser zum Durchleuchten eignen.

A. Anhang

$$L_{lum}(R, G, B) = 0.212671R + 0.71516G + 0.072169B$$

Abbildung A.3.: Formel zur Berechnung der Luminanz (aus[AMHH11], Seite 215 Gl. 7.11)

A.4. Symbolverzeichnis

(ϕ, θ)	Polarwinkel ϕ und Azimuthalwinkel θ
$l_{dist}(\phi, \theta)$	Strahldichte die in einem Punkt aus Richtung (ϕ, θ) einfällt
O	Szenenursprung, Ursprung des Weltkoordinatensystems
r_s	Szenenradius
r_l	Lichtquellenabstand zum Ursprung O
\vec{f}	Bildschirmzentrum im Raum
\vec{d}	Aufspannender Vektor der Bildschirmebene (vertikal)
\vec{r}	Aufspannender Vektor der Bildschirmebene (horizontal)
C	3×3 Kamera-Matrix (intrinsische Parameter)
D	3×4 Transformationsmatrix (extrinsische Parameter)
L	Matrix: Strahldichte die jedes Pixel in einem festgelegten Punkt erzeugt
l	Strahldichte eines Pixels, gemessen in einem bestimmten Punkt
V	Framebuffer: 8 Bit, positive Ganzzahl
v	Framebuffer: Wert eines Pixels (8 Bit)
$l_{x,y}(v)$	Ansprechverhalten des Pixels mit den Koordinaten (x, y)
$v_{x,y}(l)$	Das invertierte Ansprechverhalten des Pixels (x, y)
$l_{x,y}^0$	Kleinste Strahldichte, die von Pixel (x, y) erzeugt werden kann
$l_{x,y}^{255}$	Größte Strahldichte, die von Pixel (x, y) erzeugt werden kann
L_0	L , bei $V = 0$
L_1	L , bei $V = 1$
L_{255}	L , bei $V = 255$
L'_1	L , bei $V = 1$, wenn das Darkframe abgezogen wird
L'_{255}	L , bei $V = 255$, wenn das Darkframe abgezogen wird
L_r	Die Strahldichte einer Teilbeleuchtung
R	Dynamikbereich einer LDR-Beleuchtung
R'	Dynamikbereich einer LDR-Beleuchtung, bei Darkframesubtraktion
R_{HDR}	Dynamikbereich einer HDR-Beleuchtung
m	Länge der HDR-Sequenz
F_n	Pixelwerte (Framebuffer) des n -ten HDR-Frames
T_i	Fotografierte Szene unter der i -ten Teilbeleuchtung
D_i	Darkframe der i -ten Teilbeleuchtung
f_i	Skalierungsfaktor der i -ten Aufnahme
M	Alpha-Maske zum erzeugen der linearen Rampen am Bildschirmrand
E	Environment-Map : Zu erzeugende Umgebungsbeleuchtung
E_M	Environment-Map: Markiert die erzeugte Beleuchtung

Literaturverzeichnis

- [01] pfstools - HDR Image Manipulation Library. URL <http://pfstools.sourceforge.net>. (Zitiert auf Seite 74)
- [02] pfstmo - Tone Mapping Library. URL <http://pfstools.sourceforge.net>. (Zitiert auf Seite 74)
- [03] pfscalibration - Camera Calibration. URL <http://pfstools.sourceforge.net>. (Zitiert auf den Seiten 30, 31, 74 und 75)
- [04] High-Resolution Light Probe Image Gallery. URL <http://gl.ict.usc.edu/Data/HighResProbes>. (Zitiert auf den Seiten 14, 51 und 64)
- [05] OpenCV - Open Source Computer Vision Library. URL <http://opencv.org>. (Zitiert auf den Seiten 27 und 74)
- [06] OpenCV - Camera Calibration and 3d Reconstruction. URL http://docs.opencv.org/modules/calib3d/doc/camera_calibration_and_3d_reconstruction.html. (Zitiert auf Seite 18)
- [07] Philips Hue - Wireless LED Lamps. URL <http://meethue.com>. (Zitiert auf Seite 74)
- [08] ARToolKit - Augmented Reality Software Library. URL <http://artoolkit.sourceforge.net>. (Zitiert auf den Seiten 19, 26 und 74)
- [09] gPhoto2 - Digital Camera Software. URL <http://gphoto.sourceforge.net>. (Zitiert auf Seite 74)
- [10] OpenEXR HDR Image File Format. URL <http://www.openexr.com>. (Zitiert auf Seite 74)
- [11] dcraw - Decoding raw digital photos in linux. URL <http://www.cybercom.net/~dcoffin/dcraw>. (Zitiert auf Seite 74)
- [12] ImageMagick convert. URL <http://www.imagemagick.org/script/convert.php>. (Zitiert auf Seite 74)
- [13] GNU Compiler Collection. URL <https://gcc.gnu.org/>. (Zitiert auf Seite 74)
- [ADW04] F. Anrys, P. Dutré, Y. Willem. Image based lighting design. In *the 4th IASTED International Conference on Visualization, Imaging, and Image Processing*, S. 15. 2004. (Zitiert auf Seite 22)
- [AMHH11] T. Akenine-Möller, E. Haines, N. Hoffman. *Real-Time Rendering*. CRC Press, 3rd Auflage, 2011. (Zitiert auf den Seiten 13 und 76)

- [BN76] J. F. Blinn, M. E. Newell. Texture and Reflection in Computer Generated Images. *Commun. ACM*, 19(10):542–547, 1976. doi:10.1145/360349.360353. URL <http://doi.acm.org/10.1145/360349.360353>. (Zitiert auf Seite 12)
- [CZH⁺00] Y.-Y. Chuang, D. E. Zongker, J. Hindorff, B. Curless, D. H. Salesin, R. Szeliski. Environment Matting Extensions: Towards Higher Accuracy and Real-time Capture. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '00, S. 121–130. ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, 2000. doi:10.1145/344779.344844. URL <http://dx.doi.org/10.1145/344779.344844>. (Zitiert auf Seite 23)
- [DHT⁺00] P. Debevec, T. Hawkins, C. Tchou, H.-P. Duiker, W. Sarokin, M. Sagar. Acquiring the Reflectance Field of a Human Face. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '00, S. 145–156. ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, 2000. doi:10.1145/344779.344855. URL <http://dx.doi.org/10.1145/344779.344855>. (Zitiert auf Seite 21)
- [DM97] P. E. Debevec, J. Malik. Recovering High Dynamic Range Radiance Maps from Photographs. In *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '97, S. 369–378. ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, 1997. doi:10.1145/258734.258884. URL <http://dx.doi.org/10.1145/258734.258884>. (Zitiert auf Seite 30)
- [DMAC03] F. Drago, K. Myszkowski, T. Annen, N. Chiba. Adaptive logarithmic mapping for displaying high contrast scenes. In *Computer Graphics Forum*, Band 22, S. 419–426. Wiley Online Library, 2003. (Zitiert auf Seite 64)
- [DWT⁺02] P. Debevec, A. Wenger, C. Tchou, A. Gardner, J. Waese, T. Hawkins. A Lighting Reproduction Approach to Live-action Compositing. *ACM Trans. Graph.*, 21(3):547–556, 2002. doi:10.1145/566654.566614. URL <http://doi.acm.org/10.1145/566654.566614>. (Zitiert auf den Seiten 15, 22 und 23)
- [FBS05] M. Fuchs, V. Blanz, H.-P. Seidel. Bayesian Relighting. In *Proceedings of the Sixteenth Eurographics Conference on Rendering Techniques*, EGSR'05, S. 157–164. Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, 2005. doi:10.2312/EGWR/EGSR05/157-164. URL <http://dx.doi.org/10.2312/EGWR/EGSR05/157-164>. (Zitiert auf Seite 22)
- [LCE⁺06] B. Lamond, C.-F. Chabert, P. Einarsson, A. Jones, W.-C. Ma, T. Hawkins, M. Bolas, S. Sylwan, P. Debevec. Relighting Character Motion for Photoreal Simulations. Technischer Bericht, DTIC Document, 2006. (Zitiert auf Seite 22)
- [MDA02] V. Masselus, P. Dutré, F. Anrys. The Free-form Light Stage. In *ACM SIGGRAPH 2002 Conference Abstracts and Applications*, SIGGRAPH '02, S. 262–262. ACM, New York, NY, USA, 2002. doi:10.1145/1242073.1242275. URL <http://doi.acm.org/10.1145/1242073.1242275>. (Zitiert auf Seite 22)
- [MN99] T. Mitsunaga, S. Nayar. Radiometric self calibration. In *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on*, Band 1, S. –380 Vol. 1. 1999. doi:10.1109/CVPR.1999.786966. (Zitiert auf Seite 30)

- [MPN⁺02] W. Matusik, H. Pfister, A. Ngan, P. Beardsley, R. Ziegler, L. McMillan. Image-based 3D Photography Using Opacity Hulls. In *Proceedings of the 29th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '02, S. 427–437. ACM, New York, NY, USA, 2002. doi:10.1145/566570.566599. URL <http://doi.acm.org/10.1145/566570.566599>. (Zitiert auf Seite 23)
- [MTB⁺05] A. Mohan, J. Tumblin, B. Bodenheimer, C. Grimm, R. Bailey. Table-top Computed Lighting for Practical Digital Photography. In *Proceedings of the Sixteenth Eurographics Conference on Rendering Techniques*, EGSR'05, S. 165–172. Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, 2005. doi:10.2312/EGWR/EGSR05/165-172. URL <http://dx.doi.org/10.2312/EGWR/EGSR05/165-172>. (Zitiert auf Seite 22)
- [RBS99] M. Robertson, S. Borman, R. Stevenson. Dynamic range improvement through multiple exposures. In *Image Processing, 1999. ICIP 99. Proceedings. 1999 International Conference on*, Band 3, S. 159–163 vol.3. 1999. doi:10.1109/ICIP.1999.817091. (Zitiert auf Seite 30)
- [RBY⁺99] R. Raskar, M. S. Brown, R. Yang, W.-C. Chen, G. Welch, H. Towles, B. Seales, H. Fuchs. Multi-projector Displays Using Camera-based Registration. In *Proceedings of the Conference on Visualization '99: Celebrating Ten Years*, VIS '99, S. 161–168. IEEE Computer Society Press, Los Alamitos, CA, USA, 1999. URL <http://dl.acm.org/citation.cfm?id=319351.319370>. (Zitiert auf Seite 52)
- [RWP05] E. Reinhard, G. Ward, S. Pattanaik, P. Debevec. *High Dynamic Range Imaging: Acquisition, Display, and Image-Based Lighting (The Morgan Kaufmann Series in Computer Graphics)*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1st Auflage, 2005. (Zitiert auf den Seiten 15 und 30)
- [SB96] A. R. Smith, J. F. Blinn. Blue Screen Matting. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '96, S. 259–268. ACM, New York, NY, USA, 1996. doi:10.1145/237170.237263. URL <http://doi.acm.org/10.1145/237170.237263>. (Zitiert auf Seite 22)
- [WFZ02] Y. Wexler, A. W. Fitzgibbon, A. Zisserman. Image-based Environment Matting. In *Proceedings of the 13th Eurographics Workshop on Rendering*, EGRW '02, S. 279–290. Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, 2002. URL <http://dl.acm.org/citation.cfm?id=581896.581932>. (Zitiert auf Seite 23)
- [Zha00] Z. Zhang. A Flexible New Technique for Camera Calibration. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(11):1330–1334, 2000. doi:10.1109/34.888718. URL <http://dx.doi.org/10.1109/34.888718>. (Zitiert auf Seite 27)
- [ZWCS99] D. E. Zongker, D. M. Werner, B. Curless, D. H. Salesin. Environment Matting and Compositing. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '99, S. 205–214. ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, 1999. doi:10.1145/311535.311558. URL <http://dx.doi.org/10.1145/311535.311558>. (Zitiert auf Seite 23)

Alle URLs wurden zuletzt am 16. 6. 2014 geprüft.

Erklärung

Ich versichere, diese Arbeit selbstständig verfasst zu haben. Ich habe keine anderen als die angegebenen Quellen benutzt und alle wörtlich oder sinngemäß aus anderen Werken übernommene Aussagen als solche gekennzeichnet. Weder diese Arbeit noch wesentliche Teile daraus waren bisher Gegenstand eines anderen Prüfungsverfahrens. Ich habe diese Arbeit bisher weder teilweise noch vollständig veröffentlicht. Das elektronische Exemplar stimmt mit allen eingereichten Exemplaren überein.

Ort, Datum, Unterschrift