

Sorted Sector Covering mit Bildkondensierung

-

Eine universelle Methode zur effizienten
Berechnung von Local Dimming LED Backlight

Dissertation
zur Erlangung des Grades
des Doktors der Ingenieurwissenschaften
der Naturwissenschaftlich-Technischen Fakultät II
- Physik und Mechatronik -
der Universität des Saarlandes

von

Marc Albrecht

Saarbrücken 2010

Eingereicht am: 27.09.2010

Tag der mündlichen Prüfung: 21.03.2011

Promotionskommission

Vorsitzender: Univ.-Prof. Dr. rer. nat. Christoph Becher

Berichterstattende: Univ.-Prof. Dr.-Ing. Chihao Xu

Univ.-Prof. Dr. rer. nat. Dietrich Klakow

Prof. Dr. Dr. h.c. mult. Kurt Mehlhorn

Promovierter akad. Mitarbeiter: Dr.-Ing. Lutwin Klein

Eidesstattliche Versicherung

Hiermit versichere ich an Eides statt, dass ich die vorliegende Arbeit selbstständig und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe. Die aus anderen Quellen oder indirekt übernommenen Daten und Konzepte sind unter Angabe der Quelle gekennzeichnet. Die Arbeit wurde bisher weder im In- noch im Ausland in gleicher oder ähnlicher Form in einem Verfahren zur Erlangung eines akademischen Grades vorgelegt.

Saarbrücken, 26.09.2010

Danksagung

Ich möchte mehreren Personen, die zum Gelingen dieser Arbeit beigetragen haben, meinen Dank aussprechen.

An allererster Stelle gilt mein Dank Herrn Professor Chihao Xu, der mir die Möglichkeit gegeben hat, in seinem Team, dem Lehrstuhl für Mikroelektronik der Universität des Saarlandes, mitzuarbeiten. Professor Xu hat mir nicht nur bei der Auswahl des Themas geholfen, er gab mir zu jeder Zeit wertvolle Ratschläge, die Freiheit, eigene Ideen zu verwirklichen und, gerade zu Beginn, die notwendige Führung. Dadurch habe ich gelernt, zielgerichtet, sorgfältig und eigenständig, kurz, wissenschaftlich zu arbeiten.

Danken möchte ich, neben Herrn Professor Xu, auch der übrigen Prüfungskommission, den beiden weiteren berichterstattenden Herren Professoren Dietrich Klakow und Kurt Mehlhorn, dem Vorsitzenden Herrn Professor Christoph Becher und Herrn Dr. Lutwin Klein.

Ein besonderer Dank geht an Herrn Dr. Andreas Karrenbauer, der mit Ideen und Ratschlägen zum gelungenen Entwurf des SSC-Algorithmus beigetragen hat sowie an meinen Kollegen Herrn Tobias Jung für die fruchtbaren Gespräche, welche oftmals einen Funken der Erkenntnis im lokal gedimmten Backlight entzündet haben.

Ich möchte außerdem meinen ehemaligen Diplomanden für ihre Beiträge danken, insbesondere Herrn Daniel Schäfer für seine Arbeit zur Direct-Lit Signalanzeige und Herrn Houssam El Moutaouakil für die Hilfe bei der Analyse des Standes der Technik.

Neben der fachlichen Unterstützung war ich auch auf die Hilfe im Bereich der Administration und Verwaltung angewiesen. Ich danke daher herzlich Frau Therese Mues und Herrn Hans Morgenstern, welche einen großen Anteil daran haben, dass der Lehrstuhl in den vergangenen fünf Jahren zu einem zweiten Zuhause für mich wurde. Danke auch an alle anderen Personen, die das Leben am Lehrstuhl bereichert haben. Danke Kiso, Christoph H., Cosmin, Michael, René, Christoph D., Holger, Ayetullah, Ramazan, Bänz, Felix und alle anderen.

Mein Dank gilt auch den Personen, welche, neben meiner Familie, meine persönliche Entwicklung, bewusst oder unbewusst, positiv beeinflusst haben. Danke Michael A., Knut R., Steffen S., H. Stone, Manuel P., Felix F., Stéphane K., Torsten B., Mark B., Georg F., Jochen B., Gaetan C., Joan B. und zuletzt Jean-Marc F.

Einen großen Verdienst am Gelingen der Arbeit hat meine Frau Anna-Lena, welche mich mit einer gelungenen Mischung aus Geduld und Ansporn permanent unterstützt hat. Darüber hinaus haben Anna-Lena und mein Schwiegervater Hans, dem ich an dieser Stelle ebenfalls danken möchte, das Manuskripts Korrektur gelesen, wodurch eine nicht unbeträchtliche Zahl orthographischer und sprachlicher Fehler entdeckt und beseitigt wurden.

Zu guter Letzt gilt mein Dank meinen Eltern Annelore und Horst, meiner Schwester Kathrin und meinem Großvater Herbert für die Unterstützung in allen Lebenslagen.

Kurzzusammenfassung

Deutsch – Durch die immer größer werdenden Bildschirmdiagonalen (TV, Computermonitor) und den Wunsch nach noch mehr Flexibilität im Bereich der mobilen Kommunikation (Smartphone, Notebook, iPad) spielt der Leistungsverbrauch von Flüssigkristallanzeigen (LCDs), der zu einem Großteil vom Backlight bestimmt wird, eine immer wichtigere Rolle. Local Dimming LED Backlight ist eine Möglichkeit, den Leistungsverbrauch des Backlights um bis zu 50% zu reduzieren, indem die erzeugte Helligkeit lokal an das darzustellende Bild angepasst wird. Die Herausforderung besteht darin, möglichst viel Leistung einzusparen und gleichzeitig die Qualität des dargestellten Bildes und dessen Helligkeit zu erhalten. Um die enormen Daten eines Bildes in Echtzeitbetrieb zu verarbeiten und zugleich den Leistungsverbrauch zu minimieren, muss die Lösung effizient sein. Die bisherigen Methoden zur Berechnung von Local Dimming LED Backlight basieren auf der Bildverarbeitung und erfüllen die beschriebenen Anforderungen nur teilweise. Darüber hinaus sind sie für neue Edge-Lit Backlights ungeeignet. Im Rahmen dieser Arbeit wurde eine neuartige, auf einem mathematischen Modell des Backlights basierende, Optimierungsmethode entwickelt. Diese wurde für verschiedene Prototypen in Hardware implementiert und die daraus resultierenden Ergebnisse wurden untersucht und ausgewertet.

English - Liquid Cristal Displays (LCDs) are by far the most prevalent display technology. Growing screen sizes and mobility requirements make the reduction of the LCD's power consumption, which is mainly caused by the backlight, to one of today's hottest topics of the LCD industry. An ideal solution to reduce the power consumption to up to 50% is local dimming LED backlight. That is, the luminance of the backlight is locally adapted according to the image content. The challenge is to save as much power as possible without a degradation of the visual quality and a loss of luminance. The local dimming hardware must be quick enough for video application and as efficient as possible at the same time, for a low hardware effort. State of the Art approaches are not capable to meet all the challenges and are inapplicable for novel edge-lit backlights. That's why a new approach for the calculation of local dimming LED backlight, which is based on a mathematical model of local dimming, has been developed and presented in this work. This method has been implemented for different prototypes. The results achieved are presented and discussed.

Inhaltsverzeichnis

| | | |
|----------|--|-----------|
| 1 | EINLEITUNG | 1 |
| 1.1 | Motivation | 1 |
| 1.2 | Ziel | 2 |
| 1.3 | Aufbau dieser Arbeit | 2 |
| 2 | EINFÜHRUNG IN LCDS UND IN DEN STAND DER TECHNIK VON LOCAL DIMMING LED BACKLIGHT | 5 |
| 2.1 | Einleitung | 5 |
| 2.2 | Aufbau und Funktionsweise eines LCDs | 5 |
| 2.2.1 | Backlight eines LCDs | 8 |
| 2.2.2 | Architektur der Displaysteuerung | 11 |
| 2.3 | Stand der Technik zur Berechnung von Local Dimming LED Backlight | 14 |
| 2.3.1 | Einleitung..... | 14 |
| 2.3.2 | Algorithmus für Direct-Lit LCDs | 14 |
| 2.3.3 | Algorithmus für Edge-Lit LCDs..... | 20 |
| 2.3.4 | Fazit | 22 |
| 3 | MATHEMATISCHE FORMULIERUNG VON LOCAL DIMMING LED BACKLIGHT ... | 23 |
| 3.1 | Einleitung | 23 |
| 3.2 | Mathematisches Modell | 23 |
| 3.2.1 | Helligkeit in einem Pixel | 23 |
| 3.2.2 | Helligkeit des gesamten Backlights | 24 |
| 3.3 | Lineare Programmierung..... | 25 |
| 3.3.1 | Aufstellen eines linearen Programms | 25 |
| 3.3.2 | Lösen des linearen Programms | 26 |
| 3.4 | Fazit..... | 26 |
| 4 | SORTED-SECTOR-COVERING-ALGORITHMUS..... | 27 |
| 4.1 | Einleitung | 27 |
| 4.2 | Grundbegriffe zum Algorithmus | 27 |
| 4.2.1 | LED..... | 27 |
| 4.2.2 | Dominierende LED..... | 28 |
| 4.2.3 | Sektor | 28 |
| 4.2.4 | Partition und LED-Gruppe | 29 |
| 4.3 | Bedingung für einen Approximationsalgorithmus | 30 |
| 4.4 | Kurzüberblick zu SSC | 32 |
| 4.5 | Phase 1: Preset und Lower Bounds | 33 |
| 4.5.1 | Einleitung..... | 33 |
| 4.5.2 | Optimal Global Dimming und Preset | 33 |
| 4.5.3 | Lower Bounds..... | 34 |

| | | |
|----------|---|-----------|
| 4.6 | Phase 2: Final Duty-Cycles | 35 |
| 4.6.1 | Lokaler Optimierungsansatz | 36 |
| 4.6.2 | Globaler Optimierungsansatz | 38 |
| 4.6.3 | Parallele Prozessierung der FDC | 40 |
| 4.7 | Zwischenphase | 41 |
| 4.7.1 | IP mit globalem Optimierungsansatz | 41 |
| 4.7.2 | Festlegung des Startwertes und der Schrittweite von λ bzw. δ | 42 |
| 4.7.3 | Zusammenfassung der IP | 43 |
| 4.8 | Unterscheidung von SSC vom Stand der Technik | 43 |
| 4.9 | Merkmale von SSC | 44 |
| 4.10 | Einsatzmöglichkeiten von SSC | 45 |
| 4.11 | Güte von SSC | 46 |
| 4.11.1 | Qualität der Ergebnisse | 46 |
| 4.11.2 | Laufzeit | 48 |
| 4.12 | Fazit | 48 |
| 5 | CONDENSER – PRÄPROZESSOR FÜR SSC | 49 |
| 5.1 | Einleitung | 49 |
| 5.2 | Grundlagen des Condensers | 49 |
| 5.3 | Kondensierung der LSFs | 51 |
| 5.3.1 | Kondensierungsfunktion der LSFs | 51 |
| 5.3.2 | Bestimmung der maximalen Zellgröße der LSFs | 52 |
| 5.4 | Kondensierung der Bilddaten | 53 |
| 5.4.1 | Starre Condenser-Modi | 53 |
| 5.4.2 | Automatischer Condenser-Modus | 56 |
| 5.4.3 | Surcondenser | 60 |
| 5.5 | Fazit | 62 |
| 6 | POSTPROZESSIERUNG | 65 |
| 6.1 | Einleitung | 65 |
| 6.2 | Berechnung der TFT-Werte | 65 |
| 6.3 | Clipping | 67 |
| 6.4 | Methode zum Erhalt der Bildqualität beim Soft Clipping | 69 |
| 6.5 | Fazit | 71 |
| 7 | HW-IMPLEMENTIERUNG UND PROTOTYPEN | 73 |
| 7.1 | Einleitung | 73 |
| 7.2 | Allgemeiner Aufbau der Local Dimming Hardware | 74 |
| 7.2.1 | Speicherbausteine | 74 |
| 7.2.2 | Local-Dimming-Prozessor | 76 |
| 7.3 | Integration der Local Dimming Hardware in einen bestehenden Displaycontroller | 79 |
| 7.3.1 | Lage der Local Dimming Hardware | 79 |

| | | |
|--|--|------------|
| 7.3.2 | Pufferung und Manipulation der Bilddaten | 79 |
| 7.3.3 | Zeitlicher Ablauf der Prozessierung | 80 |
| 7.4 | Prototypen | 81 |
| 7.4.1 | Prototyp eines Direct-Lit LCDs mit 180 LEDs | 82 |
| 7.4.2 | Prototyp eines Edge-Lit LCDs mit 6 LEDs | 84 |
| 7.4.3 | Prototyp einer Signalanzeige mit 72 LEDs | 88 |
| 7.5 | Fazit | 91 |
| 8 | STATISTISCHE UND VISUELLE ERGEBNISSE | 93 |
| 8.1 | Einleitung | 93 |
| 8.2 | Statistische Ergebnisse | 93 |
| 8.2.1 | Leistungsparsnis der Prototypen | 93 |
| 8.2.2 | Qualität der Ergebnisse | 96 |
| 8.2.3 | Fazit | 100 |
| 8.3 | Visuelle Ergebnisse | 100 |
| 8.3.1 | Unbewegte Bilder | 100 |
| 8.3.2 | Video | 108 |
| 8.3.3 | Fazit | 115 |
| 9 | ZUSAMMENFASSUNG | 117 |
| 10 | AUSBLICK | 119 |
| 10.1 | 3D-Dimming | 119 |
| 10.2 | Field Sequential Color | 121 |
| 10.3 | Analyse des Zellhistogramms zur Verbesserung des automatischen Condenser-Modus | 122 |
| 10.4 | Unterdrückung von Flicker mit SSC | 124 |
| 10.5 | Stabilitätsbetrachtung | 125 |
| 10.6 | Metrik für Bildqualität | 126 |
| APPENDIX | | 127 |
| A | Gaußscher Tiefpass im Frequenzbereich | 127 |
| B | Treiben einer LED mit PWM | 128 |
| C | Bildabhängige Prozessierungsreihenfolge der Pixel in der IP | 129 |
| D | Division durch Schiebe- und Vergleichsoperationen | 131 |
| E | Pipelining | 133 |
| F | Erzeugen der LSFs | 134 |
| FORMELZEICHEN, VARIABLEN, ABKÜRZUNGEN UND GLOSSAR | | 139 |
| LITERATURVERZEICHNIS | | 147 |

1 Einleitung

1.1 Motivation

Anfangs vor allem in Digitaluhren und Taschenrechnern verwendet, sind Flüssigkristallanzeigen (LCDs) mittlerweile die mit Abstand am meisten verwendete Technologie im Displaybereich. Durch die immer größer werdenden Bildschirmdiagonalen (TV, Computermonitor) und den Wunsch nach noch mehr Flexibilität im Bereich der mobilen Kommunikation (Smartphone, Notebook, iPad) spielt der Leistungsverbrauch von LCDs, der zu einem Großteil von der Hintergrundbeleuchtung, dem Backlight, bestimmt wird, eine immer wichtigere Rolle.

In konventionellen LCDs wird mit Hilfe des Backlights eine möglichst homogene Helligkeit über die gesamte Bildfläche generiert, wodurch der Leistungsverbrauch sowohl für dunkle als auch für helle Bilder identisch hoch ist. Dadurch geht Energie unnötig verloren.

Local Dimming LED Backlight ist eine Möglichkeit, den Leistungsverbrauch erheblich zu reduzieren. Mit Hilfe von einzeln ansteuerbaren LEDs wird die Helligkeit des Backlights lokal den Anforderungen des darzustellenden Bildes angepasst. Zum besseren Verständnis ist in Abbildung 1-1 ein Beispiel für Local Dimming LED Backlight zu sehen. Dieses Beispiel zeigt das darzustellende Bild (links) und die generierte Helligkeit des lokal gedimmten Backlights (rechts), welche sich aus den überlagerten Helligkeiten der LEDs zusammensetzt.

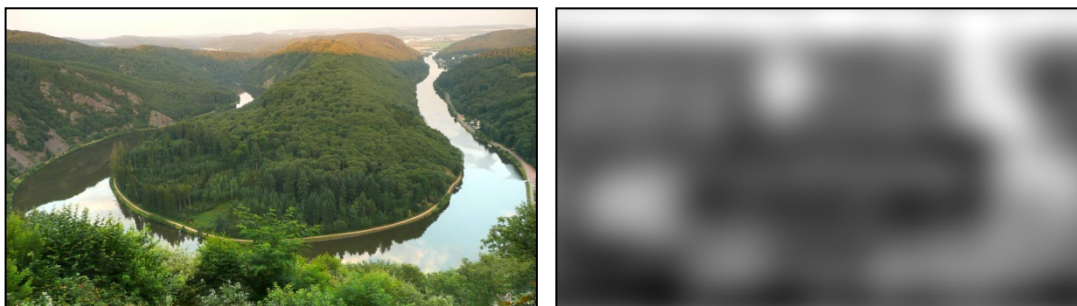


Abbildung 1-1: Beispiel für Local Dimming LED Backlight - links: Originalbild; rechts: generierte Helligkeit des lokal gedimmten Backlights

Die Herausforderung bei Local Dimming LED Backlight besteht darin, die LED-Werte so zu berechnen, dass der Leistungsverbrauch minimiert wird und gleichzeitig die visuelle Qualität und die Bildhelligkeit erhalten bleiben. Um dies erfüllen zu können, müssen die enormen Datenmengen eines Bildes im Echtzeitbetrieb verarbeitet werden, wozu es eines möglichst schnellen und zur Reduktion des Hardwareaufwandes möglichst effizienten Algorithmus bedarf.

Die Analyse bereits existierender Algorithmen zur Berechnung von Local Dimming LED Backlight ergibt, dass die prozessierten Bilder Mängel bezüglich ihrer visuellen Qualität aufweisen. Zudem eignen sich diese Algorithmen nur bedingt für neuartige Edge-Lit-Anordnungen.

1.2 Ziel

Die Berechnung der einzelnen LED-Werte in Echtzeit bedarf eines effizienten und schnellen Algorithmus. Um möglichst viel Leistung einsparen zu können, sollen dessen Ergebnisse nah an der theoretisch optimalen Lösung sein, welche sich durch den geringsten Leistungsverbrauch bei gleichbleibender visueller Qualität und Bildhelligkeit auszeichnet.

Die Entwicklung einer eigenständigen Methode zur Berechnung von Local Dimming LED Backlight, welche diese Anforderungen erfüllt, ist das Ziel der Arbeit.

1.3 Aufbau dieser Arbeit

Kapitel 2 versteht sich als Einführung in die LCD-Technik und verschafft einen Einblick in den Stand der Technik zur Berechnung von Local Dimming LED Backlight. Zunächst werden die zum Verständnis dieser Arbeit benötigten Grundlagen behandelt. Darunter fallen beispielsweise der Aufbau und die Funktionsweise eines LCDs mit besonderem Fokus auf dem Backlight. Darüber hinaus werden verschiedene Variationen des Backlights vorgestellt. Abgeschlossen wird der erste Teil des Kapitels mit der Beschreibung des Aufbaus einer Displaysteuerung.

Im zweiten Teil des Kapitels werden die Ergebnisse einer intensiven Literaturrecherche zum Stand der Technik zur Berechnung von Local Dimming präsentiert. Die publizierten Algorithmen beruhen auf bekannten Methoden der Bildverarbeitung. Exemplarisch für den Stand der Technik werden zwei publizierte algorithmische Ansätze im Detail vorgestellt und analysiert.

In Kapitel 3 wird das mathematische Modell von Local Dimming LED Backlight aufgestellt. Das Problem wird als lineares Programm formuliert und es wird gezeigt, wie dieses mit Hilfe von kommerzieller Software numerisch gelöst werden kann. Die optimalen Ergebnisse sind die Referenz für den in dieser Arbeit entwickelten Algorithmus.

Die Arbeit beinhaltet insgesamt zwei Schwerpunkte. Der erste Schwerpunkt, der entwickelte und patentierte Sorted-Sector-Covering-Algorithmus (SSC) zur Berechnung von Local Dimming LED Backlight, ist Inhalt von Kapitel 4. Die entwickelte Methode beruht, im Gegensatz zu bisher publizierten Ansätzen, auf der diskreten Mathematik. Die Ergebnisse, die mit SSC erzielt werden, weisen keinen Verlust an Bildqualität auf, sind stets sehr nah an der optimalen Lösung und benötigen nur einen geringen Aufwand an Hardware.

Der speziell für den SSC-Algorithmus entwickelte Präprozessor stellt den zweiten Schwerpunkt der Arbeit dar und ist Gegenstand von Kapitel 5. Dieser Präprozessor, der sogenannte Condenser, erlaubt eine noch höhere Energieeinsparung beim Backlight bei gleichzeitiger drastischer Reduktion der Laufzeit. Das Verhältnis der visuellen Qualität und der erzielten Leistungersparnis kann über verschiedene Condenser-Modi eingestellt werden.

Die Adaption des Backlights an das darzustellende Bild führt dazu, dass die Daten für die Ansteuerung des LCDs in einer Postprozessierung dem veränderten Helligkeitsprofil angepasst werden müssen. Daneben können hier auch Maßnahmen zur Verbesserung der visuellen Qualität vorgenommen werden. Die Vorgehensweise der Postprozessierung wird in Kapitel 6 beschrieben.

Kapitel 7 umfasst eine allgemeine Formulierung des SSC-Algorithmus, des Condensers und der Postprozessierung in Hardware. Besonderes Augenmerk wird auf eine effiziente, also eine aufwand- und kostensparende, Implementierung gelegt. Daneben wird gezeigt, wie der SSC-Algorithmus mit Condenser und Postprozessierung in eine bestehende Displaysteuerung integriert werden kann. Aufbauend auf den Ergebnissen werden dann drei Prototypen vorgestellt. Es wird demonstriert, wie die entwickelte Methode zur Berechnung von Local Dimming Backlight an die Prototypen adaptiert und in Hardware implementiert wird.

Basierend auf den Ergebnissen der Implementierung der Prototypen wird in Kapitel 8 die Leistungersparnis des SSC-Algorithmus anhand eines Test-Sets ausgewertet und mit den Ergebnissen der optimalen Lösung verglichen. Darüber hinaus wird die visuelle Qualität der prozessierten Bilder analysiert.

In Kapitel 9 werden die Ergebnisse der entwickelten Methode noch einmal kurz zusammengefasst und ein Fazit gezogen.

Ein umfassender Ausblick auf weitere Einsatzmöglichkeiten der entwickelnden Methode zur Berechnung von Local Dimming wird in Kapitel 10 gegeben.

2 Einführung in LCDs und in den Stand der Technik von Local Dimming LED Backlight

2.1 Einleitung

Dieses Kapitel stellt eine Einführung in LCDs und in den Stand der Technik von Local Dimming LED Backlight dar.

Im ersten Teil werden Aufbau und Funktionsweise von LCDs erläutert. Dabei erfolgt ein Überblick zu den verschiedenen Komponenten eines LCDs, wobei besonderes Augenmerk auf die Beschreibung der Komponente Backlight gelegt wird. Danach wird die Architektur der Ansteuerung eines LCDs beschrieben.

Im zweiten Teil wird der Stand der Technik im Bereich der Local-Dimming-Algorithmen vorgestellt. Stellvertretend für die existierenden Veröffentlichungen wird je ein Lösungsansatz für Direct-Lit und Edge-Lit Backlights im Detail erklärt, ausgewertet und diskutiert.

2.2 Aufbau und Funktionsweise eines LCDs

Abbildung 2-1 zeigt den schematischen Aufbau eines LCDs.

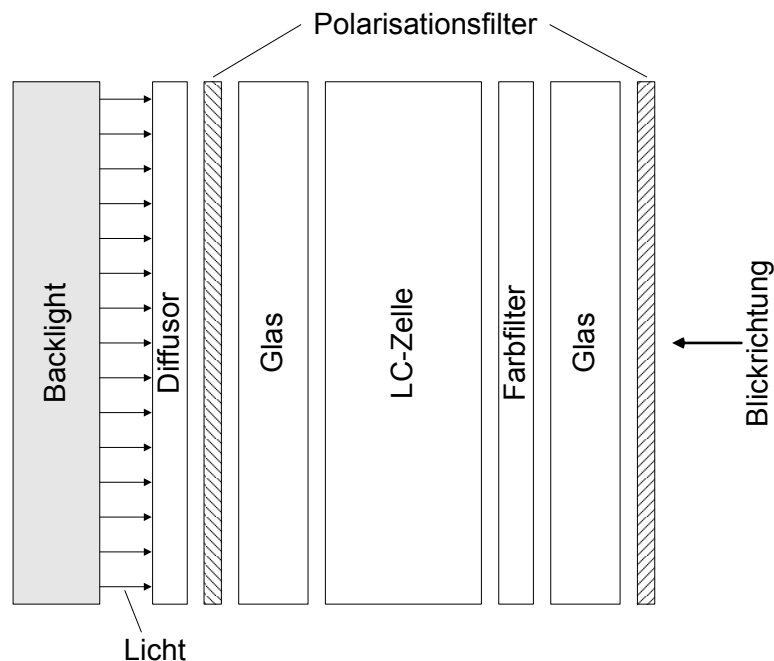


Abbildung 2-1: Schematischer Aufbau eines LCDs mit Backlight

Da dieses, im Gegensatz zu Plasma- oder OLED-Displays, nicht selbstleuchtend ist, muss die benötigte Helligkeit¹ in den Pixeln mit einer Lichtquelle, dem hier grau hinterlegten Backlight, erzeugt werden.

Die Komponenten aus Abbildung 2-1 werden nun im Einzelnen kurz vorgestellt. Eine ausführlichere Beschreibung findet sich in [Lu 01].

Diffusor

Das vom Backlight erzeugte Licht hat eine räumliche Struktur. Hinter dem Display platzierte LEDs haben eine vom Öffnungswinkel abhängige Strahlungsleistung. Die Aufgabe des Diffusors besteht, ähnlich wie bei einem Tiefpassfilter, in der Gleichverteilung der räumlichen Struktur.

Polarisationsfilter

Die nichtpolarisierten Lichtwellen, die vom Backlight generiert werden, durchlaufen sowohl beim Eintritt in die LC-Zelle als auch beim Austritt aus der LC-Zelle einen linearen Polarisationsfilter. Abbildung 2-2 zeigt die Auswirkung eines linearen Polarisationsfilters auf nichtpolarisiertes Licht.

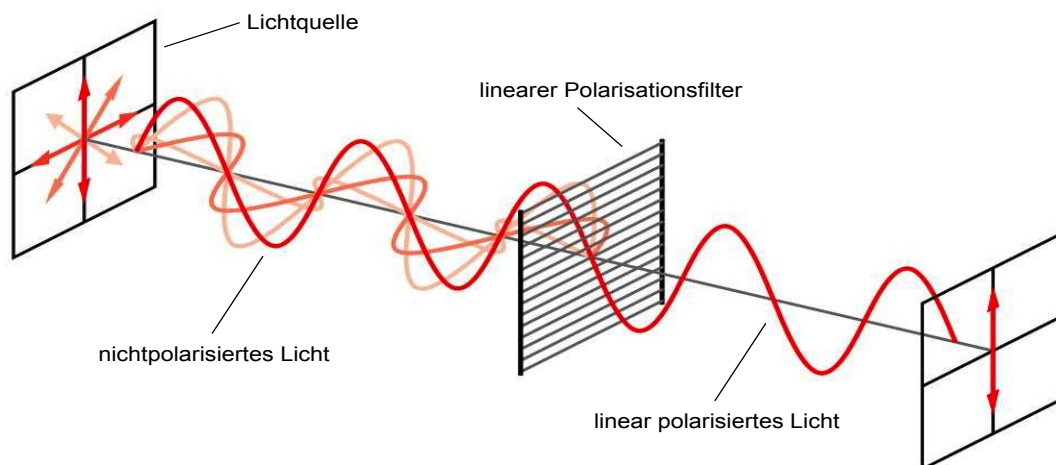


Abbildung 2-2: Grafische Veranschaulichung der linearen Polarisation (nach [HMS 97])

Der Feldvektor des austretenden Lichtes zeigt nur noch in eine Richtung und die Auslenkung ändert periodisch ihren Betrag und ihr Vorzeichen. Die beiden zum Display gehörenden Polarisationsfilter sind in der Regel um 90° zueinander verdreht angeordnet und umschließen die LC-Zelle von beiden Seiten. Der erste Polarisationsfilter wird in einschlägiger Literatur als Polarizer, der zweite als Analyzer bezeichnet.

¹ In der Displaytechnik wird üblicherweise für die Helligkeit eines Pixels der fotometrische Begriff Luminanz verwendet. In dieser Arbeit geht es jedoch darum, eine gegebene Eingangsgröße (TFT-Wert) durch Vergleichsoperationen in eine Ausgangsgröße (einen adaptierten TFT-Wert) umzuwandeln. Der physikalische Charakter der Größe spielt dabei keine Rolle. Daher wird in dieser Arbeit der Begriff Helligkeit zum besseren Verständnis verwendet.

Glas:

Die beiden Glasscheiben dienen als Schutzschicht und als Träger für die LC-Zellen.

LC-Zelle:

In der LC-Zelle wird das linear polarisierte Licht so gedreht, dass die gewünschte Helligkeit im Pixel erreicht wird. Dies geschieht durch Flüssigkristallmoleküle, die durch Anlegen einer Spannung zwischen den beiden Elektroden in ihrer Ausrichtung beeinflusst werden. Der Effekt wurde Anfang der 70er Jahre parallel von Martin Schadt und Wolfgang Helfrich [HS 70], beide Forscher der Firma Hoffmann-La Roche, sowie von James Fergason [Fer 70] von der Kent State University entdeckt und veröffentlicht. Abbildung 2-3 veranschaulicht die Funktionsweise der LC-Zelle.

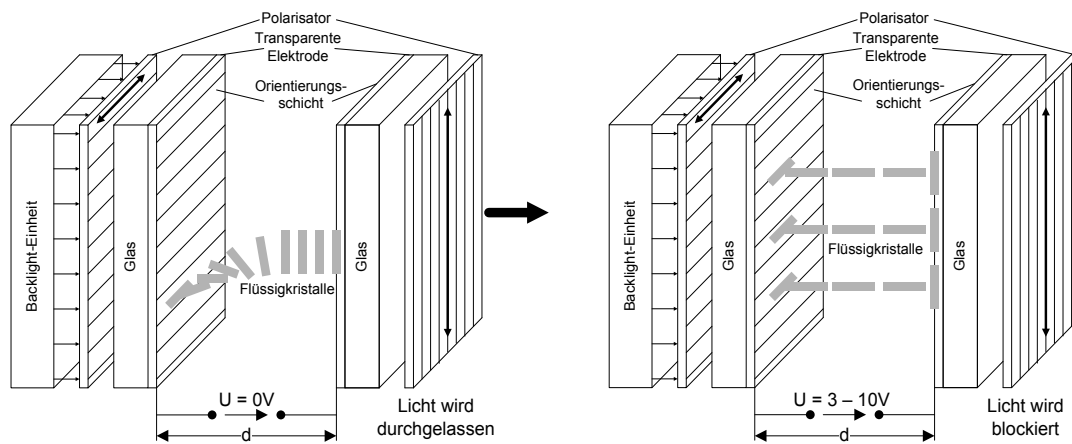


Abbildung 2-3: LC-Zelle – links: Licht wird durchgelassen; rechts: Licht wird blockiert (nach [Lu 01])

Auf der linken Seite der Abbildung 2-3 liegt keine Spannung an den Elektroden der LC-Zelle. Das linear polarisierte Licht wird von der LC-Zelle um 90° gedreht und gelangt so komplett durch den zweiten Polarisationsfilter. Der Pixel wird maximal ausgeleuchtet.

Legt man jedoch eine Spannung an die Elektroden (rechte Seite der Abbildung), wird der Drehwinkel verringert und weniger Licht gelangt durch den zweiten Polarisationsfilter. Beim Anlegen der Maximalspannung, die je nach Typ der LC-Zelle zwischen 3V und 10V liegt, wird ein Drehwinkel von nahezu 0° erreicht. In diesem Fall gelangt (fast) kein Licht durch den zweiten Polarisationsfilter.

Eine Schwäche der in Abbildung 2-3 dargestellten Twisted Nematic (TN)-Zelle ist das geringe Kontrastverhältnis, welches durch Lichtverluste (engl. light leakage) verursacht wird. Ein besseres Kontrastverhältnis bietet die weiterentwickelte Super Twisted Nematic (STN)-Zelle mit einem Drehwinkel von 180° bis 270° . Unter dem hohen Drehwinkel leidet jedoch die Farbreinheit. Abhilfe schafft hier eine zweite, nicht angesteuerte STN-Zelle oberhalb der angesteuerten STN-Zelle. Gemeinsam bilden diese beiden Zellen die Double Super Twisted Nematic (DSTN) -Zelle. An dieser Stelle sei auch noch die FSTN (Film Super Twisted

Nematic)-Zelle erwähnt, welche das Problem der Farbunreinheit dadurch löst, dass eine Farbkompensationsfolie Verwendung findet.

Eine ausführliche Beschreibung der LC-Zelltypen findet sich in [Lu 01].

Farbfilter

Ein Pixel eines farbigen LCDs setzt sich aus drei der oben beschriebenen LC-Zellen, den Subpixeln in den Farben Rot, Grün und Blau, zusammen. Aufgrund der einfachen Fertigung weist die Anordnung der Subpixel eine streifenförmige Struktur auf. Abbildung 2-4 (linke Seite) zeigt eine typische Variante.

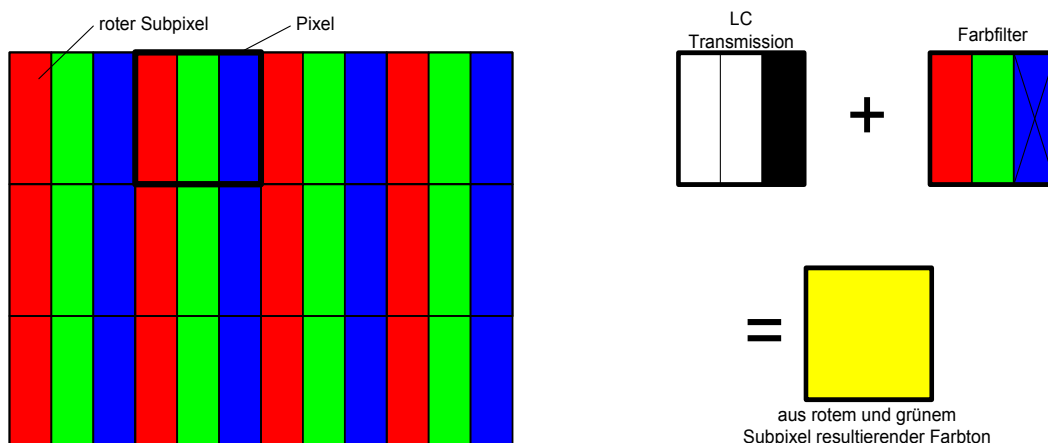


Abbildung 2-4: links: Farbfilter mit Pixel und Subpixel; rechts: Farbmischung: Additive Überlagerung von rotem und grünem Subpixel ergibt gelben Pixel

Durch die kleinen Abmessungen der Subpixel können diese vom Auge des Betrachters räumlich nicht mehr aufgelöst werden. Die wahrgenommene Farbe des Pixels entspricht einer additiven Überlagerung der drei Farbanteile der Subpixel.

Das Beispiel aus Abbildung 2-4 (rechte Seite) zeigt die Transmission der Subpixel. Der blaue Subpixel ist gesperrt, der grüne und der rote Subpixel lassen das gesamte Licht durch. Aus der Überlagerung des grünen und des roten Subpixels entsteht für den Betrachter ein gelber Pixel. Selbstverständlich kann eine LC-Zelle in der Regel neben dem kompletten Blockieren und dem Durchlassen des gesamten Lichtes auch verschiedene Zwischenpositionen einnehmen, wodurch eine größere Anzahl von Farbtönen dargestellt werden kann.

2.2.1 Backlight eines LCDs

2.2.1.1 Direktes und indirektes Backlight

Ein Großteil der aktuell erhältlichen LCDs ist mit weißem Backlight ausgestattet. Je nach Typ des LCDs und dessen Anwendungsgebiet sind die Lichtquellen an den Seiten des Displays (indirektes Backlight, engl. Edge-Lit bzw. Side-Lit) oder hinter der Anzeige (direktes Backlight, engl. Direct-Lit) angebracht. In konventionellen Displays haben beide Arten der Realisierung

zum Ziel, eine möglichst homogene Helligkeit über die gesamte Bildfläche zu erzeugen. Für eine ausführliche Beschreibung siehe [Kob 09].

Im weiteren Verlauf der Arbeit werden die beiden Begriffe Direct-Lit Backlight und Edge-Lit Backlight verwendet.

2.2.1.2 Vorteile von LEDs gegenüber CCFLs

In letzter Zeit werden die bisher in LCDs üblichen Lichtquellen, die CCFLs (Cold Cathode Fluorescent Lamp), immer häufiger durch weiße LEDs ersetzt. An der Zielsetzung des konventionellen Backlights, der Erzeugung einer möglichst homogenen Helligkeit über den gesamten Bildschirm, ändert dies nichts, jedoch liegen durch stetige Verbesserungen die Vorteile mittlerweile klar auf Seiten der LED:

- Der Wirkungsgrad, sprich der Quotient aus abgegebenem Lichtstrom und aufgenommener Leistung, von LEDs ist mittlerweile höher als der von CCFLs [Ana 08].
- LEDs haben darüber hinaus eine längere Lebensdauer als CCFLs und sind robuster in Bezug auf mechanische und thermische Beanspruchung.
- Des Weiteren benötigen sie eine weniger aufwändige Treiberschaltung, da sie im Gegensatz zu CCFLs, welche auf mehrere hundert Volt Versorgungsspannung angewiesen sind, mit wenigen Volt auskommen.
- Auch die EMV-Problematik, die bei der Verwendung von CCFLs entsteht, entfällt.
- Durch die Verwendung von Pulsweitenmodulation (PWM) besteht bei LEDs die Möglichkeit, die Helligkeit der angesteuerten LED in nahezu beliebig feinen Abstufungen einzustellen. Die Dimmbarkeit von CCFLs beträgt dagegen maximal 20%.
- LEDs verfügen über eine sehr kurze Reaktionszeit (100ns). Die Reaktionszeit von CCFLs beträgt mehrere Millisekunden (siehe [Ana 08]).
- Ein weiterer Vorteil von weißen LEDs gegenüber weißen CCFLs wird beim Betrachten der Abbildung 2-5 deutlich. Der darstellbare Farbraum der weißen LEDs ist größer. Durch das Mehr an Farben wird die Qualität der dargestellten Bilder verbessert.

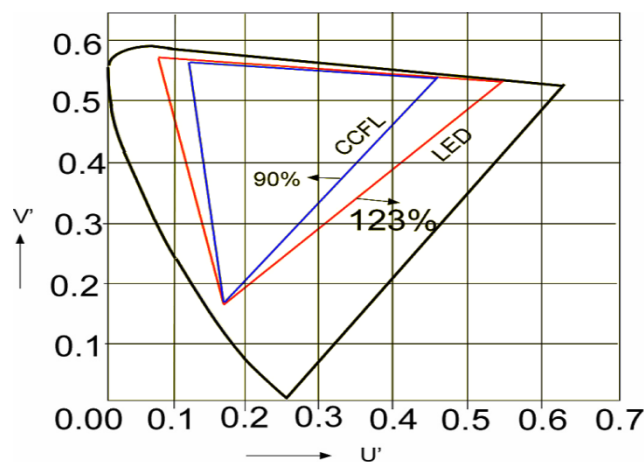


Abbildung 2-5: Vergleich des darstellbaren Farbraums von weißen CCFLs mit dem von weißen LEDs [Ana 08]

- Die Abmessungen von LEDs erlauben LCDs mit geringerer Einbautiefe. Dadurch sind sie auch in mobilen Anwendungen wie Notebooks oder Handys von Vorteil, wo dank der einstellbaren Farbtemperatur der LEDs das Backlight perfekt an die Umgebung angepasst werden kann.
- Darüber hinaus wird zur Herstellung von LEDs kein gesundheitsschädliches Quecksilber benötigt.
- Auch erlaubt die LED fortschrittliche Techniken, wie beispielsweise Field Sequential Color (siehe Abschnitt 10.2), welche mit CCFLs nicht realisierbar sind.

2.2.1.3 Dimmen des Backlights

Der größte Nachteil bei der Verwendung von konventionellen Backlights, unabhängig von der verwendeten Lichtquelle, ist, dass stets überall die gleiche Helligkeit erzeugt wird. Es spielt keine Rolle, ob das darzustellende Bild dunkel oder hell ist. Der Leistungsverbrauch des Backlights ist somit unabhängig vom Bildinhalt, obwohl bei dunklen Bildern durch Dimmen des Backlights Leistung eingespart werden könnte.

Generell unterscheidet man drei Arten des Dimmens, welche im Folgenden kurz erläutert werden.

Globales Dimmen (0D-Dimming, Global Dimming):

Der naheliegende Ansatz, bei dem das Backlight einheitlich gedimmt wird, wird globales Dimmen (0D-Dimming, Global Dimming) genannt. Globales Dimmen kann, mit gewissen Einschränkungen (geringe maximale Dimmbarkeit, lange Reaktionszeit), auch mit CCFLs erreicht werden.

Eine Möglichkeit global zu dimmen wäre, den hellsten Pixel des darzustellenden Bildes zu detektieren und das Backlight so einzustellen, dass der hellste Pixel noch ausreichend ausgeleuchtet wird.

Eindimensionales Dimmen (1D-Dimming):

Beim eindimensionalen Dimmen (1D-Dimming) wird das Backlight streifenweise (horizontal bzw. vertikal) gedimmt. Auch hier können theoretisch, nimmt man die bereits erwähnten Einschränkungen in Kauf, CCFLs verwendet werden. Allerdings ist dem Autor kein kommerzielles Produkt dieser Art bekannt. Es ist leicht einzusehen, dass diese Art des Dimmens ein höheres Potential an Leistungersparnis hat als das 0D-Dimming.

Lokales Dimmen (2D-Dimming, Local Dimming):

Der fortschrittlichste und effektivste Ansatz wird lokales Dimmen (2D-Dimming, Local Dimming) genannt. Bei einem Direct-Lit Backlight erreicht man dies, indem man die LEDs in einem gitterförmigen Raster platziert und einzeln bzw. in kleinen Gruppen ansteuert. Dadurch ist es möglich, die Helligkeit des Backlights lokal an das darzustellende Bild anzupassen.

Die drei hier vorgestellten Arten des Dimmens sind grafisch vereinfacht in Abbildung 2-6 dargestellt.

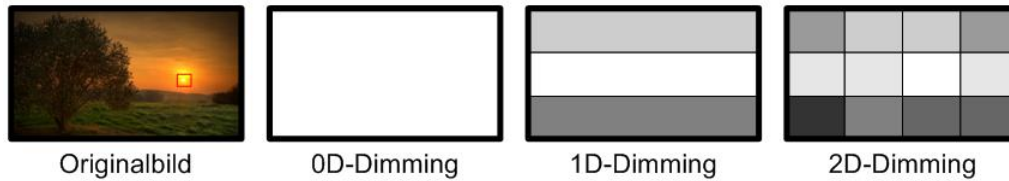


Abbildung 2-6: Grafischer Vergleich von 0D-, 1D- und 2D-Dimming

Auf der linken Seite der Abbildung sieht man das darzustellende Bild. Der hellste Pixel des Bildes (rote Markierung) bestimmt den Wert des global gedimmten Backlights. Bei der Verwendung von 1D-Dimming (hier mit drei unterschiedlich ansteuerbaren Lichtquellen) kann die Helligkeit des Backlights besonders im unteren, dunklen Teil des Bildes reduziert werden. Die größte Helligkeitsreduktion – und somit die höchste Leistungseinsparung – ist mit lokalem Dimmen zu erreichen, hier mit 12 einzeln ansteuerbaren Lichtquellen.

2.2.2 Architektur der Displaysteuerung

Nachdem der Aufbau und die Funktionsweise eines LCDs und die verschiedenen Backlight-Typen eingeführt wurden, wird nun die Architektur der Ansteuerung eines LCDs beschrieben. Diese besteht aus den in Abbildung 2-7 gezeigten Blöcken Videoprozessor und LCD-Modul, die im Anschluss näher ausgeführt werden.

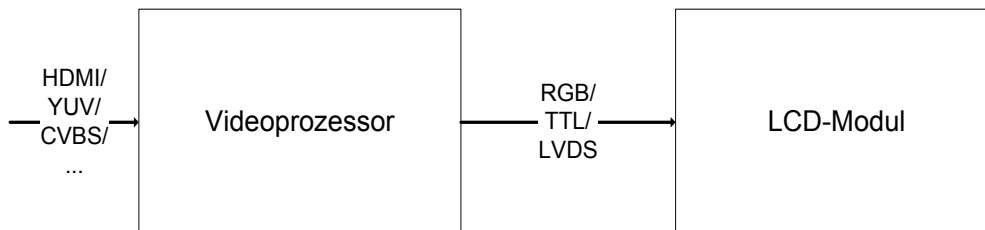


Abbildung 2-7: Architektur der Displaysteuerung

2.2.2.1 Videoprozessor

Der Videoprozessor hat die Aufgabe, das Quellsignal in ein für das LCD-Modul kompatibles Signal umzuwandeln. Neben der Anpassung der Bildwiederholrate und der Auflösung können hier auch Maßnahmen zur Verbesserung der Bildqualität getroffen werden. Abbildung 2-8 zeigt einen möglichen Aufbau des Videoprozessors. Die darin enthaltenen Komponenten werden im Einzelnen kurz vorgestellt.

Quelle:

Über die Schnittstelle mit der Umgebung werden die Quelldaten an den Videoprozessor herangeführt. Als Quelle kommen beispielsweise ein TV-Receiver, ein DVD-Player etc. in Frage.

Front-End-Prozessierung:

Das von der Bildquelle kommende Signal wird für die weitere Verarbeitung vorbereitet. So werden hier beispielsweise DVD-Daten dekodiert.

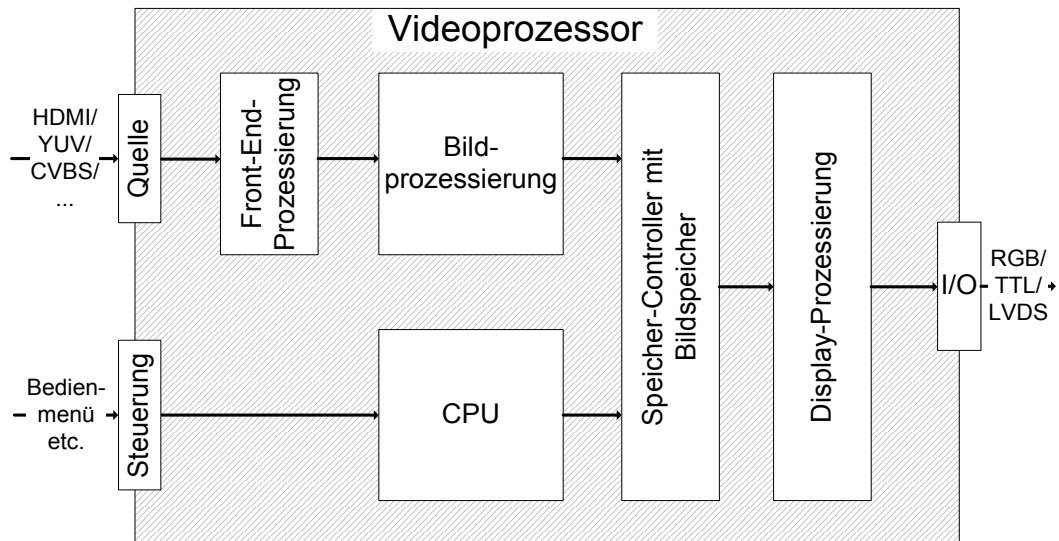


Abbildung 2-8: Aufbau eines Videoprozessors

Bildprozessierung:

Die vorbereiteten Daten aus der Front-End-Prozessierung werden bei der Bildprozessierung aufgearbeitet. Hierunter fällt, neben dem Entfernen von Rauschen und dem Deinterlacing, vor allem die Skalierung des Bildes auf die vom Display unterstützte Auflösung. Des Weiteren erfolgt hier auch die Erhöhung der Framerate von beispielsweise 30 Frames/Sekunde auf 120 Frames/Sekunde.

CPU:

Die Aufgabe der CPU ist es, die vom Benutzer gemachten Eingaben auszuführen. Gleichzeitig wird der Speicher-Controller gesteuert.

Speicher-Controller mit Bildspeicher:

Die aufgearbeiteten Daten werden in den Bildspeicher geladen. Der Speicher-Controller wird von der CPU gesteuert.

Steuerung:

Über diese Schnittstelle kann der Benutzer beispielsweise das Kontrastverhältnis oder die Helligkeit des Displays beeinflussen.

Display-Prozessierung

Die sich im Bildspeicher befindenden Daten werden in die Display-Prozessierung eingelesen. Hier werden Maßnahmen zur Verbesserung der Bildqualität durchgeführt.

Als Resultat erhält man diskrete Werte für jeden Pixel bzw. Subpixel des Displays. Diese werden im weiteren Verlauf dieser Arbeit als Pixelwerte oder TFT-Werte bezeichnet. Die Werte und verschiedene Synchronisationssignale werden schließlich an das LCD-Modul weitergegeben.

2.2.2.2 LCD-Modul

Mit den vom Videoprozessor generierten TFT-Werten werden im LCD-Modul die einzelnen LC-Zellen angesteuert und so ein Bild angezeigt. Abbildung 2-9 zeigt das Blockdiagramm des LCD-Moduls. Die einzelnen Komponenten werden im Anschluss erläutert.

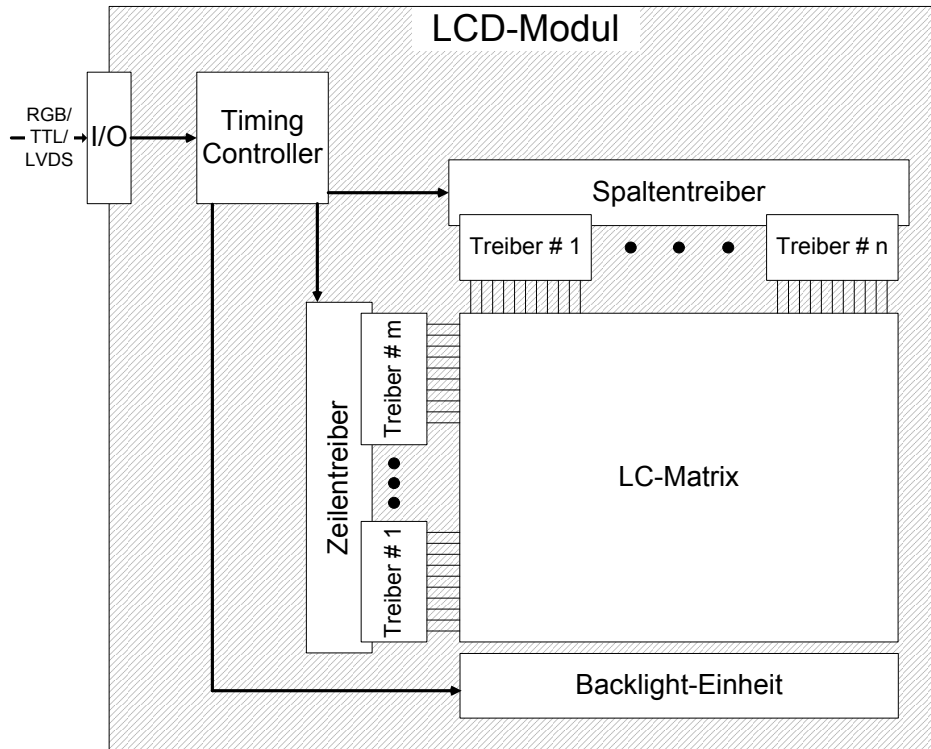


Abbildung 2-9: LCD-Modul

Timing Controller:

Bei konventionellen Displays ist es die Aufgabe des Timing Controllers, die Zeilen- und Spaltentreiber zu synchronisieren. Bei Displays mit Local Dimming wird das Backlight für jedes darzustellende Bild neu angepasst. Daher müssen hier nicht nur die Zeilen und Spalten miteinander synchronisiert werden, sondern diese auch mit dem Backlight.

Zeilen- und Spaltentreiber:

Über Synchronisationssignale vom Timing Controller wird das Bild pixelweise, von oben links nach unten rechts, aufgebaut. Dabei werden die digital kodierte und vom Videoprozessor erzeugte TFT-Werte in analoge Signale umgewandelt, um die einzelnen LC-Zellen ansteuern und somit deren Transmission bestimmen zu können.

Je nachdem, wie die einzelnen Pixel angesprochen werden, unterscheidet man zwischen Aktiv- und Passivmatrixadressierung (AM bzw. PM). PM ist billiger in der Herstellung, allerdings beschränkt in der Auflösung. Daher werden moderne LCDs in der Regel mit AM realisiert. In den meisten AM-Displays finden die sogenannten Dünnschichttransistoren (TFTs) Verwendung, wobei für jeden Subpixel ein eigener TFT zum Einsatz kommt. Aus diesem Grund werden in dieser Arbeit die Pixelwerte auch als TFT-Werte bezeichnet.

Für eine ausführlichere Beschreibung von AM und PM siehe [Lu 01]. Der Aufbau und die Funktionsweise von TFTs werden in [Lee 08] detailliert erläutert.

LC-Matrix:

Die LC-Matrix setzt sich aus den LC-Zellen zusammen. Diese werden über die Zeilen- und Spaltentreiber so eingestellt, dass für den Betrachter das gewünschte Bild entsteht.

Backlight-Einheit:

Die Backlight-Einheit besteht aus den Lichtquellen und deren Treibern. Beim Local Dimming erhalten die Treiber die mit dem Bild synchronisierten LED-Werte vom Timing Controller.

2.3 Stand der Technik zur Berechnung von Local Dimming LED Backlight

2.3.1 Einleitung

Mittlerweile bieten viele der großen TV-Hersteller, wie beispielsweise Samsung, Sony und Philips, TV-Geräte mit Local Dimming Direct-Lit Backlight an. In den letzten Jahren erschienen zahlreiche Veröffentlichungen von Algorithmen zur Berechnung von Local Dimming LED Backlight (siehe [An 08], [Ch 06], [Ch 07], [CKK 08], [GH 07], [HLL 08], [Lu 08], [Pen 07], [Se 04], [Sh 08], [Shi 07], [Yeo 08]). Im Gegensatz dazu gibt es nur eine Veröffentlichung für das lokale Dimmen von Edge-Lit LCDs [HG 09].

Die grundlegende Vorgehensweise hinter diesen Ansätzen ist dieselbe. Das darzustellende Bild mit hohem Informationsgehalt wird auf die geringe Auflösung des Backlights skaliert, um die LEDs dann auf einen durch Mittelung der Pixelwerte festgelegten Wert einzustellen.

Exemplarisch für die verschiedenen Lösungsansätze für Direct-Lit LCDs wird der im Oktober 2006 auf der ASID-Konferenz in Indien [Ch 06] und, in einer weiterentwickelten Version, im Oktober 2007 im Journal der SID [Ch 07] vorgestellte Lösungsansatz von Samsung präsentiert, analysiert und diskutiert. Dieser Ansatz wurde in Hardware implementiert und die Ergebnisse an einem Prototyp ausgewertet. Mittlerweile wird er in Local Dimming TVs von Samsung verwendet. Im Anschluss daran wird der algorithmische Lösungsansatz von NXP [HG 09] für die Berechnung von Local Dimming für Edge-Lit LCDs vorgestellt und bewertet.

2.3.2 Algorithmus für Direct-Lit LCDs

Die Leuchtdioden werden beim Algorithmus für Direct-Lit LCDs zu einzeln ansteuerbaren, in Rechtecken angeordneten Strängen zusammengefasst. Im Folgenden wird die Bezeichnung LED für solch einen Strang von Leuchtdioden verwendet (siehe auch Abbildung 4-1). Das darzustellende Bild wird, orientiert an der Position der LEDs, ebenfalls in kleinere, nicht überlappende Teilbilder zerlegt.

Der Algorithmus besteht, wie in Abbildung 2-10 dargestellt, aus mehreren Teilschritten, welche in den nun folgenden Abschnitten detailliert erläutert werden.

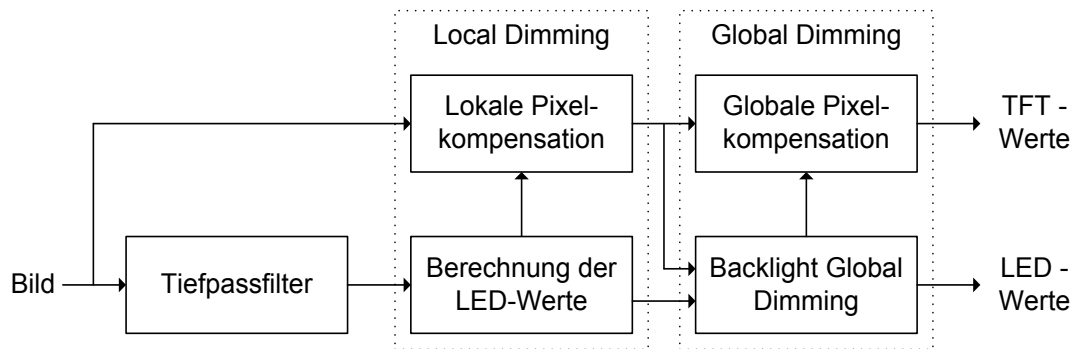


Abbildung 2-10: Ablaufdiagramm des Algorithmus für Direct-Lit LCDs

2.3.2.1 Tiefpassfilter

Ziel dieses ersten Schrittes ist es, die hohe optische Auflösung des darzustellenden Bildes auf die geringe optische Auflösung des LED-Rasters zu reduzieren. Dies wird mit Hilfe eines Tiefpassfilters erreicht.

Da in der Veröffentlichung selbst keine Aussage über die verwendete Filterfunktion gemacht wird, wurde zur Nachbildung des Algorithmus ein Gaußsches Tiefpassfilter im Frequenzbereich (siehe Appendix A – Gaußscher Tiefpass im Frequenzbereich) verwendet. Der Vorteil gegenüber einem Tiefpassfilter im Zeitbereich ist der bedeutend geringere Rechenaufwand.

Das Resultat der Tiefpassfilterung ist in Abbildung 2-11 in einem Beispielbild zu sehen.

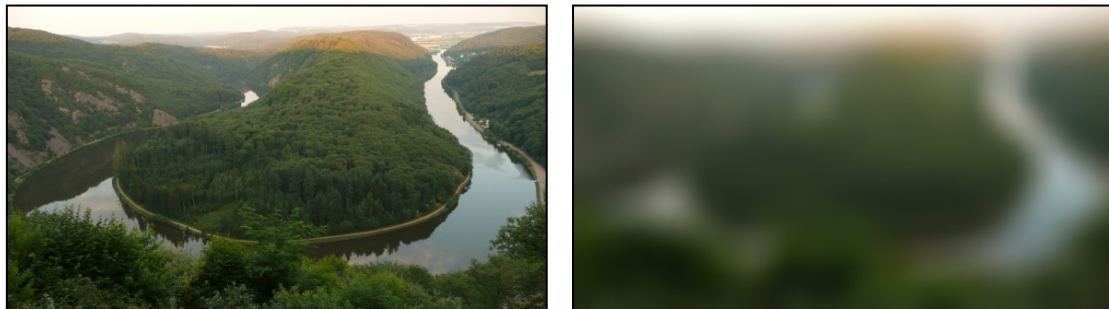


Abbildung 2-11: Vergleich zwischen Original (links) und dem entsprechenden tiefpassgefilterten Bild (rechts)

Auf der linken Seite der Abbildung befindet sich das Originalbild, auf der rechten Seite das mit einem Gaußschen Tiefpass gefilterte Bild. Klare Konturen der Bildgegenstände sind beim tiefpassgefilterten Bild nicht mehr zu erkennen, der Informationsgehalt ist erheblich reduziert.

2.3.2.2 Local Dimming

Der zweite Schritt des Algorithmus unterteilt sich in die Berechnung der LED-Werte und in die lokale Pixelkompensation.

Berechnung der LED-Werte

Die Berechnung der LED-Werte erfolgt in drei Schritten (siehe Abbildung 2-12).

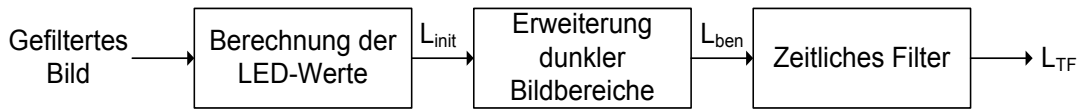


Abbildung 2-12: Ablauf der Berechnung der LED-Werte

Zunächst werden die Werte $L_{init}(m,n)$ der einzelnen LEDs mit Hilfe der Formel 2.1 festgelegt.²

$$L_{init}(m,n) = \min\left(L_{MAX}, \sum_{i=0}^{255} (H_{(m,n)}(i) \cdot W(i))\right) \quad 2.1$$

Die Indizes (m,n) bezeichnen den Ort der aktuell betrachteten LED in der m -ten Reihe und n -ten Spalte der LED-Matrix, der Index (i) steht für den Pixelwert des jeweils betrachteten Pixels. Die LED-Werte errechnen sich aus dem Minimum des maximal möglichen LED-Wertes L_{MAX} und einem Wert, der sich aus dem Histogramm $H_{(m,n)}(i)$ des betrachteten Bildausschnittes und einer Gewichtungsfunktion $W(i)$ ergibt. Je größer dabei der Wert des betrachteten Pixels ist, desto größer ist die Gewichtung.

Um zu vermeiden, dass kleine, in einen dunklen Hintergrund eingebettete, helle Strukturen verloren gehen, wird ein Schritt zur Erweiterung dunkler Bildbereiche vollzogen. Hierzu wird zuerst der arithmetische Mittelwert L_{mean} der Pixelwerte des betrachteten Bildausschnittes berechnet. Dann werden die LED-Werte L_{ben} nach der folgenden Formel modifiziert:

$$L_{ben} = \begin{cases} L_{init} & \text{if } (L_{init} < L_{mean}) \text{ or } (L_{mean} > T_M) \\ \min(L_{MAX}, L_{init} + T_B(L_{init} - L_{mean})) & \text{else} \end{cases} \quad 2.2$$

Der Parameter T_M ist ein Schwellwert, mit dessen Hilfe festgestellt wird, ob es sich bei dem aktuell betrachteten Bildausschnitt um einen dunklen Bildausschnitt handelt. Mit Hilfe des Parameters T_B wird der Grad der Pixelwerterweiterung festgelegt. Wenn es sich nicht um einen dunklen Teilabschnitt handelt, dann gilt $L_{ben} = L_{init}$.

Abgeschlossen wird die Berechnung der LED-Werte mit dem zeitlichen Filter, welches zur Aufgabe hat, Flicker (siehe auch Abschnitt 10.4) zu unterdrücken. Hierzu wird ein an den Bildinhalt adaptives IIR-Filter eingesetzt, um die LED-Werte aufeinanderfolgender Frames³ anzugleichen (siehe Formel 2.3).

$$L_{TF}^{(k)}(m,n) = R \cdot L_{ben}^{(k)}(m,n) + (1-R) \cdot L_{ben}^{(k-1)}(m,n) \quad 2.3$$

² Hinweis: Die in diesem Abschnitt verwendeten Formelzeichen wurden aus [Ch 07] übernommen. Diese stimmen nicht mit den im weiteren Verlauf der Arbeit eingeführten Formelzeichen überein.

³ Ein einzelnes Bild aus einer Videosequenz wird allgemein als Frame bezeichnet.

Dieses Filter ermittelt die einzelnen LED-Werte L_{TF} für Frame (k) aus dem entsprechenden L_{ben} vom aktuellen und vom vorangegangenen Frame. Die Aufgabe des Parameters R ($0 < R < 1$) ist es, Flicker beim Übergang von einem Frame zum nächsten Frame zu unterdrücken. R berechnet sich analog zu Formel 2.4.

$$R = \min\left(1, T_{TF} + \left|P_{mean}^{(k)} - P_{mean}^{(k-1)}\right|\right) \quad 2.4$$

P_{mean} ist hier der durchschnittliche Pixelwert des aktuellen (k) bzw. des vorangegangenen Frames ($k-1$). Der Parameter T_{TF} wurde von den Entwicklern empirisch ermittelt. Wenn sich der durchschnittliche Pixelwert P_{mean} zweier aufeinanderfolgender Frames stark unterscheidet, ist die Wahrscheinlichkeit groß, dass es sich um einen Bildschnitt handelt. Dies führt zu einem hohen Wert von R . Das Backlight ändert sich also vor allem dann schnell, wenn ein Szenenschnitt eintritt, ansonsten wird eine langsame Veränderung der LED-Werte angestrebt.

Lokale Pixelkompensation

Der zweite Teilschritt bei der Berechnung von Local Dimming ist die lokale Pixelkompensation. Hier werden die TFT-Werte an die aktuelle Helligkeit des Backlights angepasst.

Zuerst wird die von den LED-Werten $L(m,n)$ hervorgerufene Helligkeit $B(i,j)$ des Backlights ermittelt. Die hierfür benötigten Helligkeitsverteilungen $f_{m,n}(i,j)$, welche den Einfluss der einzelnen LEDs auf alle Pixel des Bildes beschreiben, sind in Form einer LUT hinterlegt (siehe Formel 2.5).

$$B(i, j) = \sum_{m,n} [f_{m,n}(i, j) \cdot L(m, n)] \quad 2.5$$

Im nächsten Teilschritt des Algorithmus wird ein Kompensationsfaktor $C(i,j)$ für jeden Pixel festgelegt. Dieser beruht auf dem Verhältnis zwischen TFT-Wert und durchgelassener Helligkeit im Pixel. Eine genaue Beschreibung dieses Verhältnisses erfolgt in Abschnitt 6.2.

Die folgende Gleichung gibt das Verhältnis zwischen dem Kompensationsfaktor $C(i,j)$ (> 1), dem erreichten Helligkeitswert im Pixel $P(i,j)$ und der geforderten Helligkeit im Pixel $I(i,j)$ wieder.

$$C(i, j) = 1 + \text{LUT}[P(i, j), I(i, j)] \quad 2.6$$

Je größer die Differenz zwischen der erreichten Helligkeit und der Sollhelligkeit in einem Pixel ist, desto kleiner wird der Kompensationsfaktor.

Falls das Backlight aus weißen LEDs besteht, ist der Kompensationsfaktor für alle drei farbigen Subpixel identisch. Im Fall von RGB-LEDs gibt es für jeden Subpixel einen eigenen Kompensationsfaktor. Die RGB-Kanäle des darzustellenden Bildes werden abschließend mit dem zuvor errechneten Kompensationsfaktor erweitert. Bei weißen LEDs werden alle

Farbkanäle identisch behandelt (siehe Formel 2.7). Verwendet man RGB-LEDs, gilt laut den Autoren Formel 2.8.

$$\{R'(i, j), G'(i, j), B'(i, j)\} = C(i, j) \cdot \{R(i, j), G(i, j), B(i, j)\} \quad 2.7$$

$$\begin{aligned} R'(i, j) &= C_R(i, j) \cdot R(i, j) \\ G'(i, j) &= C_G(i, j) \cdot G(i, j) \\ B'(i, j) &= C_B(i, j) \cdot B(i, j) \end{aligned} \quad 2.8$$

Damit ist der zweite Schritt des Algorithmus (Local Dimming) abgeschlossen. Es wurden die einzelnen LED-Werte festgelegt und das Ursprungsbild Pixel für Pixel der berechneten Helligkeit des Backlights angepasst.

2.3.2.3 Global Dimming

Der letzte Schritt des Algorithmus beinhaltet die beiden Teilschritte Backlight Global Dimming und globale Pixelkompensation.

Der Fokus beim Backlight Global Dimming liegt auf der Reduktion von Lichtverlusten. Während das Augenmerk beim zuvor beschriebenen Local Dimming auf den dunklen Bildbereichen liegt, wird nun versucht, das gesamte Backlight global zu dimmen. Dabei soll möglichst wenig Bildinformation verloren gehen.

Hierzu wird auf Basis des Histogramms des darzustellenden Bildes ein kritischer Pixelwert A ($0 \leq A \leq \text{maximaler Pixelwert}$) anhand eines Schwellwertes T_{GD} ermittelt. Dieser Schwellwert gibt die erlaubte Pixelzahl oberhalb des kritischen Pixelwertes vor. Alle Pixel, die heller sind als A , werden als Rauschen betrachtet. Zur genauen Ermittlung des Schwellwertes wird keine Aussage gemacht. Je kleiner der Schwellwert gewählt wird, desto mehr Leistung kann eingespart werden. Allerdings steigt damit auch der Verlust an Informationsgehalt. Nach der Bestimmung des kritischen Pixelwertes werden alle LEDs um einen Faktor $(A/255)^{\gamma}$ gedimmt. Gleichzeitig werden die Pixelwerte mit dem Faktor $(255/A)$ multipliziert, wobei alle Werte, die größer als A sind, auf den maximalen Pixelwert gesetzt werden.

Abbildung 2-13 veranschaulicht die Vorgehensweise grafisch.

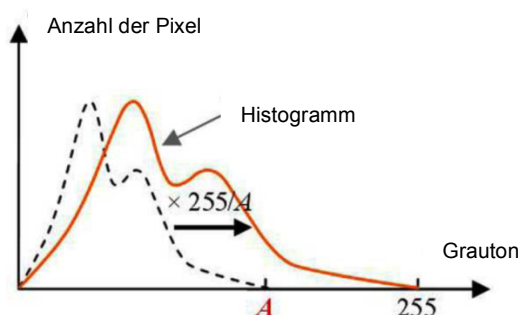


Abbildung 2-13: Vorgehensweise beim Global Dimming: Streckung des Histogramms (nach [Ch 07])

Dieser Vorgang entspricht einer gleichmäßigen Streckung der Pixelwerte. Die dabei entstehenden Rundungsfehler werden vernachlässigt.

2.3.2.4 Bewertung des Direct-Lit-Ansatzes

Der Lösungsansatz von Samsung ist in der Lage, Local Dimming LED Backlight in Echtzeit zu berechnen. Über die Komplexität der Hardware zum Erhalten der LED-Werte wird nur gesagt, dass das fertige Design auf einen FPGA passt.

Die einzige aufwändige Berechnung ist die Tiefpassfilterung im ersten Schritt der Prozessierung. Alle anderen mathematischen Operationen können, auch durch die Verwendung verschiedener LUTs, mit überschaubarem Aufwand bewältigt werden.

Der Hauptkritikpunkt, der sich bei der genauen Analyse des Algorithmus aufdrängt, ist die schlechte Qualität der prozessierten Bilder. Dadurch, dass die Intensitäten der LEDs an ein mit einem Tiefpass gefiltertes Bild – ein Bild mit einem geringen Informationsgehalt - adaptiert werden, besteht die Gefahr, dass die erreichte Helligkeit unter der Sollhelligkeit liegt. Dieses Phänomen wird Clipping genannt (siehe Abschnitt 6.3). Je nachdem, wie konzentriert das Clipping auftritt, droht sogar der Verlust wichtiger Bilddetails, da einzelne, helle Bereiche nicht ausgeleuchtet werden. Abbildung 2-14 zeigt auf der linken Seite einen vergrößerten Bildausschnitt von Abbildung 2-11.



Abbildung 2-14: links: betrachteter Bildausschnitt; Mitte: vergrößerte Darstellung des hellen Objekts; rechts: gleicher Bildausschnitt gefiltert mit einem Tiefpass

Das helle Objekt im Originalbild ist nach der Tiefpassfilterung nicht mehr zu sehen und wird somit bei der Berechnung der LED-Werte nicht weiter berücksichtigt. Durch die zusätzliche Mittelwertbildung (siehe Formel 2.1 und 2.2) wird dieser Effekt noch weiter verstärkt. Es besteht die Gefahr, dass dieses Objekt im prozessierten Bild später entweder gar nicht mehr oder zumindest stark verdunkelt auftauchen wird.

Beim Betrachten des nächsten Beispiels (siehe Abbildung 2-15) wird der beschriebene Effekt auf drastische Weise sichtbar:

Das angezeigte Bild des konventionellen LCDs auf der linken Seite von Abbildung 2-15 unterscheidet sich stark vom angezeigten Bild des Local-Dimming-Prototyps auf der rechten Seite. Viel Information ist verloren gegangen. Nicht nur, dass einzelne Sterne nicht angezeigt werden, auch größere Objekte, wie beispielsweise der Sternennebel, sind nicht mehr erkennbar.

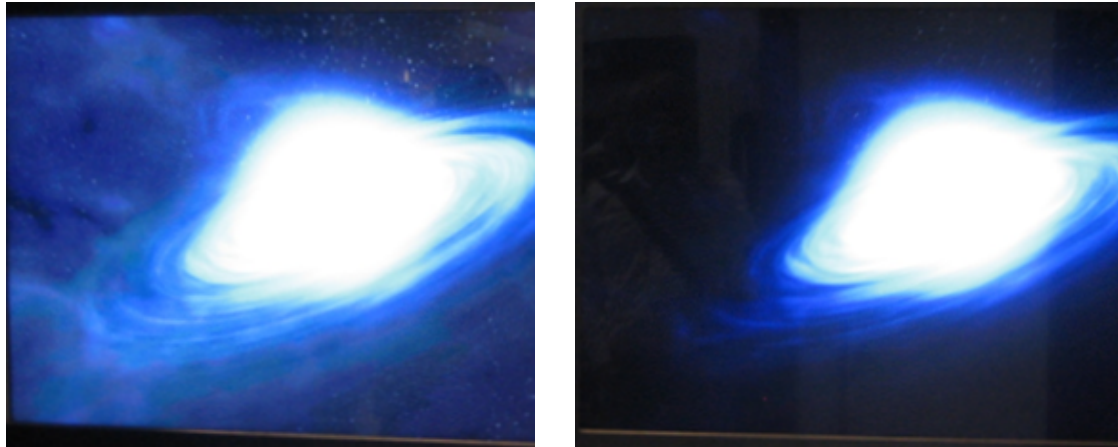


Abbildung 2-15 (selbst gemachte Fotos während der Ausstellung zur SID 2009 in San Antonio, USA) - links: Konventionelles LCD von Samsung; rechts: Local-Dimming-Prototyp von Samsung

Abhilfe würde eine Modifikation in den ersten beiden Schritten des Algorithmus schaffen. Durch Entfernen des Tiefpassfilters und Festlegen von L_{init} (siehe Formel 2.1) auf jeweils den maximalen Wert des sich im betrachteten Gebiet befindenden Pixels könnte man Ergebnisse frei von Clipping erhalten.

Hier fällt dann ein weiterer Schwachpunkt des Algorithmus auf. Das Übersprechen der LEDs wird nicht bei der Bestimmung der LED-Werte berücksichtigt. Das verringert die Komplexität, führt aber zu schlechten Ergebnissen, da zu viel Helligkeit generiert wird.

Durch das globale Dimmen im letzten Schritt des Algorithmus wird zwar Leistung eingespart, es kann jedoch viel Helligkeit im Vergleich zum ursprünglichen Bild verloren gehen. Dadurch ist es schwierig, eine Aussage bezüglich der tatsächlichen Leistungersparnis zu machen.

Ein letzter Kritikpunkt zeigt sich bei der genauen Betrachtung von Formel 2.8. Hier werden die resultierenden Pixelwerte der lokalen Pixelkompensation bei Verwendung von farbigen LEDs berechnet. Die Berechnung der einzelnen Farbkanäle erfolgt dabei unabhängig voneinander, der Farb-Crosstalk wird nicht berücksichtigt. Dies kann zu Farbverschiebungen und somit ebenfalls zu einer Verschlechterung der Bildqualität führen. Diese Tatsache wird im Ausblick dieser Arbeit genauer erläutert (siehe Abschnitt 10.1).

2.3.3 Algorithmus für Edge-Lit LCDs

In diesem Abschnitt wird der algorithmische Lösungsansatz von NXP [HG 09] für die Berechnung von Local Dimming für Edge-Lit LCDs vorgestellt.

Bei einem Edge-Lit LCD werden ebenfalls mehrere Dioden zu einer LED zusammengefasst. Die Dioden sind in einem Strang angeordnet. Das in der Publikation gezeigte LCD hat jeweils fünf LEDs an der Ober- und Unterseite des Displays.

Im Anschluss an die Beschreibung der Segmentierung des Displays wird der eigentliche Algorithmus dargelegt, welcher aus zwei Phasen besteht: Der Bestimmung der LED-Werte und der Adaption der TFT-Werte.

2.3.3.1 Segmentierung des Displays

Für jedes Display werden die Pixel in verschiedene Segmente unterteilt. Die Anzahl und die Position der Segmente werden von der Anzahl und der Position der LEDs bestimmt. Abbildung 2-16 zeigt die Unterteilung eines Displays in 10 Segmente und den Querschnitt der Helligkeitsverteilung einer LED.

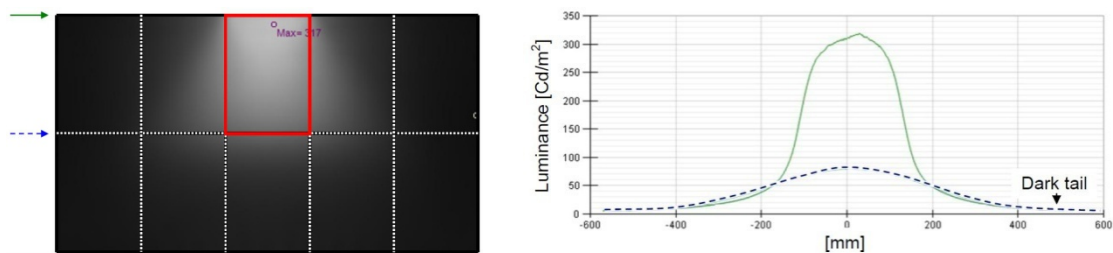


Abbildung 2-16: links: Segmentierung des Displays mit Helligkeitsverteilung einer LED; rechts: Querschnitt des Helligkeitsprofils (aus [HG 09])

2.3.3.2 Bestimmung der LED-Werte

Die einzelnen LED-Werte werden aus einem Mittelwert aller sich im Segment befindenden Pixel bestimmt. Die genaue Berechnungsvorschrift wird in der Publikation nicht offengelegt. Man kann jedoch davon ausgehen, dass es sich um eine ähnliche Vorgehensweise handelt, wie sie bereits in Abschnitt 2.3.2.2 beschrieben wurde.

2.3.3.3 Berechnung der TFT-Werte

Basierend auf der Helligkeitsverteilung der LEDs und den ermittelten LED-Werten wird die erreichte Helligkeit in jedem Pixel berechnet. Gepaart mit den Pixelwerten können daraus die TFT-Werte ermittelt werden. Es werden keine genauen Aussagen zur Vorgehensweise gemacht, jedoch ist auch hier zu erwarten, dass es sich um eine ähnliche Vorgehensweise wie in Abschnitt 2.3.2.2 handelt.

2.3.3.4 Bewertung des Edge-Lit-Algorithmus

Mit dem beschriebenen Lösungsansatz ist NXP in der Lage, Local Dimming in Echtzeit zu berechnen. Jedoch weist auch diese Methode einige Mängel auf, welche im Folgenden dargelegt werden.

Wie in Abbildung 2-16 zu sehen ist, beeinflusst eine LED nicht nur das ihr zugeordnete Segment, sondern auch benachbarte Segmente. Dies wird jedoch bei der Bestimmung der LED-Werte nicht berücksichtigt, wodurch mehr Helligkeit erzeugt und somit mehr Leistung als notwendig verbraucht wird.

Des Weiteren kann, je nach Art der Mittelwertbildung, viel Bildinformation verloren gehen. Das dadurch entstehende Clipping sollte jedoch weniger schwer ins Gewicht fallen als beim Direct-Lit-Ansatz, da eine LED beim Edge-Lit Backlight mehr Pixel beeinflusst. Das nicht berücksichtigte Übersprechen der LEDs kann somit einen größeren Beitrag zur Helligkeit leisten.

Eine wesentliche Schwäche ist, dass die Methode nicht für alle Edge-Lit Backlights geeignet ist. Es ist nicht möglich, ein Backlight mit LEDs an drei oder vier Seiten des Displays auf die beschriebene Art zu segmentieren, da ein Segment dann von mehreren LEDs beeinflusst wird.

2.3.4 Fazit

Der Stand der Technik liefert Ergebnisse zur Berechnung von Local Dimming LED Backlight in Echtzeit. Jedoch genügt keines der beiden präsentierten Verfahren den Anforderungen, die an diese Arbeit gestellt werden, da die Ergebnisse Mängel bezüglich ihrer visuellen Qualität aufweisen. Mit den beschriebenen Methoden ist es nicht möglich, Ergebnisse frei von Clipping zu erzielen, zumindest nicht nahe an der jeweiligen optimalen Lösung des Bildes, da ein Übersprechen der LEDs beim Festlegen der LED-Werte nicht berücksichtigt wird.

Im Gegensatz zu Direct-Lit Local Dimming existiert auf dem Gebiet von Edge-Lit Local Dimming aktuell nur die oben gezeigte Methode, welche eingeschränkt in ihrer Anwendung ist und keine zufriedenstellenden Ergebnisse liefert.

Es ist jedoch bekannt, dass alle großen Hersteller auf diesem Gebiet tätig sind und in absehbarer Zeit erste Produkte mit Edge-Lit Local Dimming auf den Markt kommen sollen, da LCDs mit Edge-Lit Backlight einen entscheidenden Vorteil haben. Die Anzahl der verbauten Leuchtdioden ist geringer. So benötigt ein Direct-Lit LCD mit einer Bilddiagonale von 32“ etwa 1000 Leuchtdioden, ein gleich großes Edge-Lit LCD dagegen nur etwa 200. Damit sind die Herstellungskosten von Edge-Lit LCDs erheblich geringer. Es ist offensichtlich, dass ein effizienter Local-Dimming-Algorithmus für Edge-Lit LCDs, welcher Ergebnisse nahe der optimalen Lösung bei gleichbleibender visueller Qualität liefert, von großem Nutzen ist.

3 Mathematische Formulierung von Local Dimming LED Backlight

3.1 Einleitung

Der in Abschnitt 2.3 präsentierte Stand der Technik zur Berechnung von Local Dimming beruht auf gängigen Methoden der Bildverarbeitung. Die damit erzielten Ergebnisse bieten jedoch noch Raum für Verbesserungen. Aus diesem Grund wird eine neuartige Vorgehensweise angestrebt. Diese stützt sich auf ein im Rahmen dieser Arbeit entwickeltes mathematisches Modell des Backlights, welches auf der Lichterzeugung, der Verteilung des Lichtes und Lichtfiltern (LC-Zellen) basiert.

Zunächst wird gezeigt, wie die erreichte Helligkeit in einem Pixel berechnet wird, welche sich aus der eingestellten Intensität des Backlights und dem Einfluss des Backlights auf den Pixel ergibt. Darauf aufbauend wird das mathematische Modell von Local Dimming Backlight entwickelt. Das Problem der Berechnung von Local Dimming wird, basierend auf diesem Modell, in einem linearen Programm formuliert. Es wird gezeigt, wie dieses lineare Programm mit der kommerziellen Software CPLEX optimal gelöst werden kann.

3.2 Mathematisches Modell

3.2.1 Helligkeit in einem Pixel

LEDs, egal ob weiß oder farbig, sind nicht-kohärente Lichtquellen. Eine Überlagerung der Helligkeit zweier nicht-kohärenter Lichtquellen entspricht mathematisch einer Addition der Helligkeiten.

Möchte man die Helligkeit b , die in einem Pixel p erreicht wird, berechnen, muss man die in diesem Pixel erzeugten Helligkeiten der einzelnen LEDs summieren. Diese setzen sich wiederum aus den Einflüssen d_p der entsprechenden LEDs im betrachteten Pixel multipliziert mit den Intensitäten der L einzelnen LEDs $x'(\ell)$ zusammen (siehe Formel 3.1).

$$b_p = \sum_{\ell=1}^L [d_p(\ell) \cdot x'(\ell)] \quad 3.1$$

Der Einfluss einer LED auf einen Pixel hängt im Wesentlichen vom Abstand des Pixels zur LED ab. Je weiter ein Pixel von einer LED entfernt ist, desto kleiner ist auch der Einfluss, den die LED auf ihn ausübt.

Da die ausgestrahlte Farbe einer LED abhängig von der Stromstärke ist, ist es notwendig, die LEDs immer mit einer konstanten Stromamplitude zu betreiben. Die Variation der Helligkeit erreicht man durch Pulsweitenmodulation (PWM - siehe Appendix B - Treiben einer LED mit Pulsweitenmodulation). Dadurch kann man die Intensitäten $x'(\ell)$ der LEDs aus Formel 3.1 durch PWM-Werte $x(\ell)$ ersetzen. In Formel 3.1 steht a_p , analog zu d_p , für den Einfluss der LED auf den entsprechenden Pixel, allerdings nun normiert auf den PWM-Wert der zugeordneten LED.

$$b_p = \sum_{\ell=1}^L [a_p(\ell) \cdot x(\ell)] \quad 3.2$$

Der jeweilige PWM-Wert $x(\ell)$ liegt zwischen 0 und 1 und repräsentiert die Zeit, in der die ℓ -te LED mit einer festgelegten Helligkeit leuchtet. Ist $x(\ell)$ gleich null, ist die LED die ganze Zeit ausgeschaltet, ist $x(\ell)$ gleich eins, leuchtet diese die ganze Zeit.

3.2.2 Helligkeit des gesamten Backlights

Weitet man Formel 3.2 auf alle Pixel des Bildes aus, erhält man das folgende System von Gleichungen:

$$\begin{bmatrix} a_{1,1} & a_{1,1} & \dots & a_{1,L} \\ a_{2,1} & a_{2,2} & & \\ \dots & & \dots & \\ a_{p,1} & & & a_{p,L} \end{bmatrix} * \begin{bmatrix} x_1 \\ x_2 \\ \dots \\ x_L \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \dots \\ b_P \end{bmatrix} \quad 3.3$$

Die Einflussmatrix A setzt sich dabei aus den Einflüssen aller LEDs auf alle Pixel des Bildes zusammen, wobei $a_{p,\ell}$ den Einfluss der ℓ -ten LED auf den p -ten Pixel beschreibt. Die Anzahl der Zeilen von A entspricht der Anzahl der Pixel des Displays P ⁴. Die Anzahl der Spalten von A entspricht der Anzahl der getrennt voneinander ansteuerbaren LEDs L . Die erreichte Helligkeit b wird, wie in Formel 3.2, durch Summierung der LED-Helligkeiten im Pixel berechnet.

Das in Formel 3.3 aufgestellte Gleichungssystem beschreibt den Zusammenhang zwischen dem Einfluss der LEDs auf die Pixel, den LED-Werten und der daraus resultierenden Helligkeit in den Pixeln. Möchte man nun ein Bild mit der geforderten Helligkeit in jedem Pixel versorgen,

⁴ Da es sich um ein Modell für Local Dimming LED Backlight handelt, welches die Verwendung von weißen Lichtquellen impliziert, wird nur jeweils der hellste der Subpixel eines Pixels bei der Berechnung berücksichtigt.

so müssen die LED-Werte groß genug sein, um die geforderte Helligkeit in jedem Pixel abdecken zu können.

3.3 Lineare Programmierung

3.3.1 Aufstellen eines linearen Programms

Erweitert man Formel 3.3 um das darzustellende Bild i so ergibt sich das folgende System von Ungleichungen:

$$\begin{bmatrix} a_{1,1} & a_{1,1} & \dots & a_{1,L} \\ a_{2,1} & a_{2,2} & & \\ \dots & & \dots & \\ a_{P,1} & & & a_{P,L} \end{bmatrix} * \begin{bmatrix} x_1 \\ x_2 \\ \dots \\ x_L \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \dots \\ b_P \end{bmatrix} \geq \begin{bmatrix} i_1 \\ i_2 \\ \dots \\ i_P \end{bmatrix} \quad 3.4$$

Ausgehend von einem Backlight, bei welchem jeder Pixel ausreichend ausgeleuchtet wird, wenn alle LEDs mit ihrem maximalen Wert betrieben werden, hat dieses Ungleichungssystem mindestens eine Lösung.

Die geforderte Nebenbedingung, dass das erzielte Ergebnis frei von Clipping sein muss, sprich, dass die erreichte Helligkeit b in jedem Pixel mindestens so groß wie die vom Bild geforderte Helligkeit i ist, wird durch $b \geq i$ sichergestellt.

Der Überschuss an Helligkeit in einem Pixel wird von der jeweiligen LC-Zelle kompensiert. Es soll möglichst viel Licht durch die LC-Zelle durchgelassen werden, so dass das Backlight so weit wie möglich abgedunkelt werden kann.

Das eigentliche Ziel von Local Dimming ist, dass die Verlustleistung des Backlights minimiert wird. Da die LEDs mit PWM betrieben werden, ist es einleuchtend, dass eine Minimierung der Verlustleistung P_v gleichbedeutend mit der Summe der PWM-Werte ist:

$$P_v \propto \sum_{\ell=1}^L x(\ell) \quad 3.5$$

Das in Formel 3.4 aufgestellte System von Ungleichungen, gepaart mit den aufgestellten Bedingungen, ergibt das folgende lineare Programm (LP, siehe auch [BT 97] und [Co 2002]):

$$\min \left\{ \sum_{\ell=1}^L x(\ell) : Ax \geq i, \quad 0 \leq x \leq 1 \right\} \quad 3.6$$

Ziel ist es also, die Summe der PWM-Werte zu minimieren und gleichzeitig zu gewährleisten, dass die erreichte Helligkeit in jedem Pixel mindestens der vom Bild geforderten Helligkeit entspricht ($Ax \geq i$) und dass der PWM-Wert $x(\ell)$ für jede LED einen gültigen Wert hat.

Gültig bedeutet in diesem Fall, dass der PWM-Wert zwischen null (LED ist ausgeschaltet) und eins (LED leuchtet über die gesamte Zeit) liegt.

Aufgrund der Zielsetzung, die Leistungersparnis zu maximieren, wird ein weiterer möglicher Vorteil von Local Dimming, die Erhöhung des statischen Kontrastes, nicht bei der Berechnung berücksichtigt.

3.3.2 Lösen des linearen Programms

Zur Bestimmung der optimalen Lösung des linearen Programms aus Formel 3.6 wird im Rahmen dieser Arbeit die Software CPLEX [ILOG 10] verwendet. CPLEX benutzt zum Lösen eines solchen Problems den Simplex Algorithmus. Dieser stellt das gegebene lineare Programm durch einen mehrdimensionalen Polyeder dar. Geometrisch gesprochen wandert der Algorithmus die Polyederkanten solange ab, bis er an einer optimalen Ecke angelangt ist.

Die genaue Funktionsweise dieses Algorithmus ist nicht Gegenstand dieser Arbeit, wird jedoch in [Cor 02] und [DA 98] ausführlich dargelegt.

Die mit CPLEX berechneten optimalen Lösungen werden in dieser Arbeit als Referenz verwendet.

3.4 Fazit

Mit dem präsentierten mathematischen Modell wird das Problem der Berechnung von Local Dimming von der Bildverarbeitung in die lineare Optimierung überführt. Das lineare Programm kann optimal mit CPLEX gelöst werden. Jedoch benötigt CPLEX für ein Full-HD Display mit 180 LEDs mehrere Minuten Berechnungszeit pro Bild auf einem Rechnercluster. Es steht somit außer Frage, dass solch eine numerische, optimale Lösung sowohl zu kosten- und rechenintensiv als auch zu langsam für eine Implementierung in Hardware ist.

Daher ist es notwendig, einen effizienten Approximationsalgorithmus für die Berechnung von Local Dimming zu entwickeln. Dieser wird im folgenden Kapitel präsentiert.

4 Sorted-Sector-Covering-Algorithmus

4.1 Einleitung

Wie in Abschnitt 3.4 bereits erwähnt, ist eine numerische Lösung von Local Dimming Backlight sehr aufwändig und zeitintensiv und somit in der Praxis ungeeignet. Der Stand der Technik im Bereich der Local-Dimming-Backlight-Algorithmen (siehe Abschnitt 2.3) lässt Raum für Verbesserungen.

Die eigenständig entwickelte Methode mit dem Namen Sorted Sector Covering (SSC) ist Gegenstand dieses Kapitels. Hierbei handelt es sich um den ersten von zwei Schwerpunkten dieser Arbeit.

Es erfolgt zunächst die Definition verschiedener für den Algorithmus verwendeter Begriffe. Dann wird gezeigt, dass die Einflussmatrix A dünn besetzt ist, womit die Bedingung für einen Approximationsalgorithmus erfüllt ist. Die verschiedenen Phasen von SSC werden anschließend im Detail offengelegt. Darauf folgen eine Abgrenzung von SSC vom Stand der Technik, eine Auflistung der wesentlichen Merkmale sowie eine Beschreibung der Einsatzmöglichkeiten von SSC. Abgeschlossen wird dieses Kapitel durch eine Betrachtung der Güte des Approximationsalgorithmus und ein Fazit.

4.2 Grundbegriffe zum Algorithmus

4.2.1 LED

Wenn im Laufe der Arbeit der Begriff LED verwendet wird, dann bezeichnet dieser, wenn nicht anders angegeben, eine Reihenschaltung von mehreren Leuchtdioden, welche alle vom selben elektrischen Signal angesteuert werden und somit ein (nahezu) identisches Verhalten zeigen.

Beim Direct-Lit Backlight sind die einzelnen Dioden in Form eines Gitters angeordnet (siehe Abbildung 4-1).

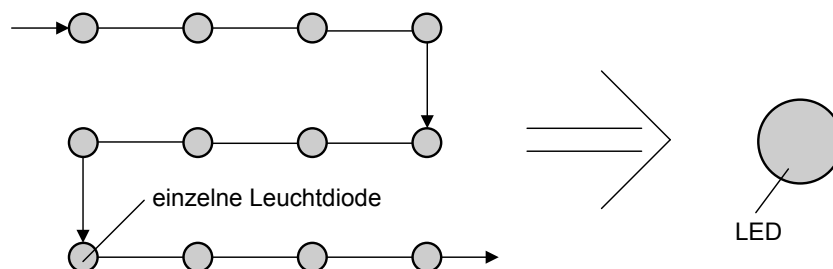


Abbildung 4-1: Reihenschaltung einzelner Leuchtdioden (links) und das Ersatzschaltbild einer LED (rechts)

Beim Edge-Lit Backlight sind die einzelnen Dioden in einer Reihe an einer Seite oder auch mehreren Seiten des Displays platziert. Im Beispiel aus Abbildung 4-3 sind die sechs LEDs des Backlights an allen vier Seiten des Displays angebracht – je zwei oben und unten und je eine an der Seite.

4.2.2 Dominierende LED

Ein Pixel wird von einer LED dominiert, wenn diese LED den größten Einfluss auf den Pixel hat. Da der Einfluss einer LED auf einen Pixel mit größer werdendem Abstand zur LED abnimmt, hat dies zur Folge, dass diejenige LED, welche den dominanten Einfluss auf einen Pixel hat, gleichzeitig die geringste Entfernung zu diesem Pixel aufweist.

Falls ein Pixel derart zwischen mehreren LEDs liegt, dass diese einen ähnlichen Einfluss auf den Pixel ausüben, so kann der Pixel auch gleichzeitig von mehreren LEDs dominiert werden.

4.2.3 Sektor

Alle Pixel, welche von derselben LED bzw. denselben LEDs dominiert werden, bilden einen Sektor. Bei Direct-Lit Backlights haben solche Sektoren wegen der gitterförmigen Anordnung der LEDs im Allgemeinen die Form eines Rechtecks (siehe linke Seite der Abbildung 4-2).

Bei einem Edge-Lit Display variiert die Form je nach Position des Sektors und Verteilung der LEDs. Ein Beispiel hierfür ist auf der rechten Seite von Abbildung 4-2 zu sehen.

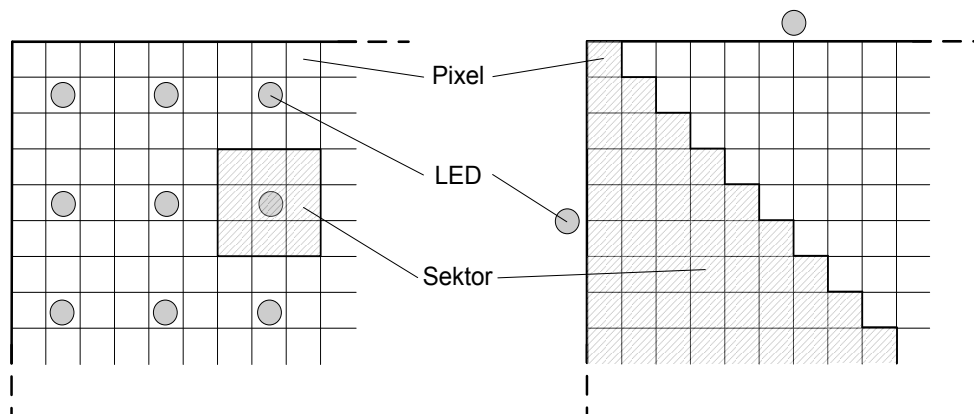


Abbildung 4-2: Sektoren eines LCDs - links: Direct-Lit; rechts: Edge-Lit

Auf die Pixel nahe dem Übergang von einem Sektor zu einem anderen Sektor ist der Einfluss der dominierenden LEDs nahezu identisch. In diesem Fall besteht, wie bereits oben erwähnt, die Möglichkeit, diese Pixel mehreren LEDs zuzuordnen. Abbildung 4-3 zeigt dies am Beispiel eines Edge-Lit Displays. Die hierfür verwendeten Daten entstammen den LSFs aus Abbildung 4-7.

Die LEDs 1 bis 6 sind an den Seiten des Displays angebracht. Die linke Seite der Abbildung 4-3 zeigt die sechs Sektoren, wenn man jedem Pixel eine einzige dominierende LED zuordnet. Im Beispiel auf der rechten Seite wurden Pixel, auf die eine oder mehrere LEDs nahezu identischen

Einfluss haben, den entsprechenden LEDs zugeordnet. Die weiß hinterlegten Bereiche sind Sektoren, die unmittelbar der jeweiligen LED zugeordnet sind. Die hellgrau eingefärbten Sektoren werden von zwei LEDs dominiert. Bei den dunkelgrauen Bereichen haben drei LEDs einen ähnlichen Einfluss auf die Pixel. Der schwarz eingefärbte Sektor in der Mitte des Displays wird von allen sechs LEDs in etwa gleich beeinflusst.

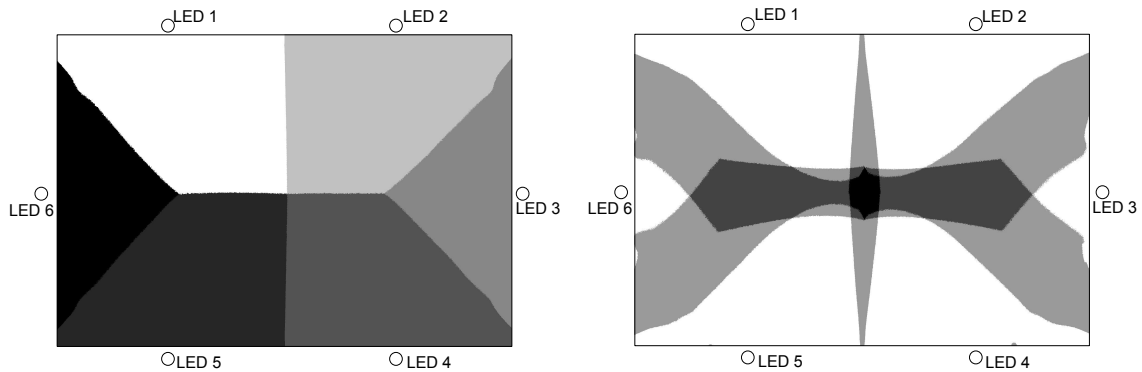


Abbildung 4-3: Sektoren eines Edge-Lit Displays – links: Sektor ist einer LED zugeordnet; rechts: Sektor ist einer oder mehreren LEDs zugeordnet

4.2.4 Partition und LED-Gruppe

Zur Reduktion der Komplexität ist es möglich, das Display in Partitionen zu unterteilen. In Abbildung 4-4 wird ein Beispiel für Direct-Lit Backlight gezeigt. An den Ecken jeder Partition sitzt hier, bis auf die Partitionen im unmittelbaren Randbereich des Displays, je eine LED. Gemeinsam bilden diese LEDs eine LED-Gruppe.

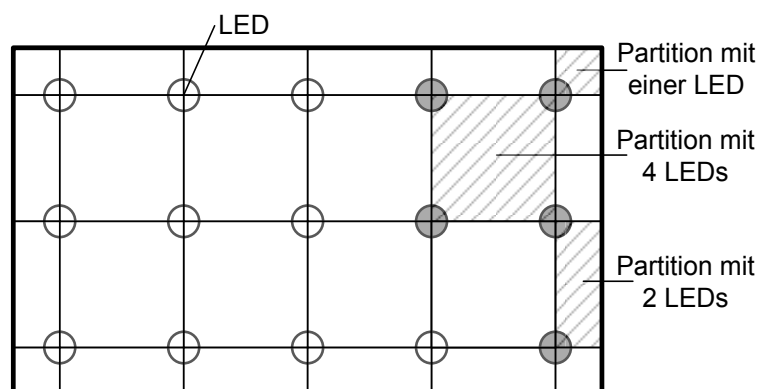


Abbildung 4-4: Grafische Erläuterung des Begriffs Partition

Im Folgenden wird auf die grau eingefärbten LEDs eingegangen. In den Ecken besteht die LED-Gruppe aus einer LED, an den Seiten aus zwei und in der Mitte aus vier LEDs. Man erkennt, dass eine LED mehreren Partitionen und somit auch mehreren LED-Gruppen zugeordnet werden kann. Die Partitionen mit den zugeordneten LED-Gruppen werden unabhängig voneinander betrachtet. Zur Berechnung eines Pixels werden nur die Einflüsse der zugeordneten LED-Gruppe berücksichtigt.

Es ist offensichtlich, dass das Vernachlässigen der Beiträge der nicht zur LED-Gruppe gehörenden LEDs zu einer Verschlechterung der Ergebnisse führen kann. Aus diesem Grund ist es möglich, weitere LEDs zur Berechnung der Pixel einer Partition mit einzubeziehen. Abbildung 4-5 veranschaulicht dies anhand eines Beispiels. Hier werden nicht nur die umrandenden LEDs der LED-Gruppe zugeordnet, sondern ebenfalls weiter entfernte LEDs.

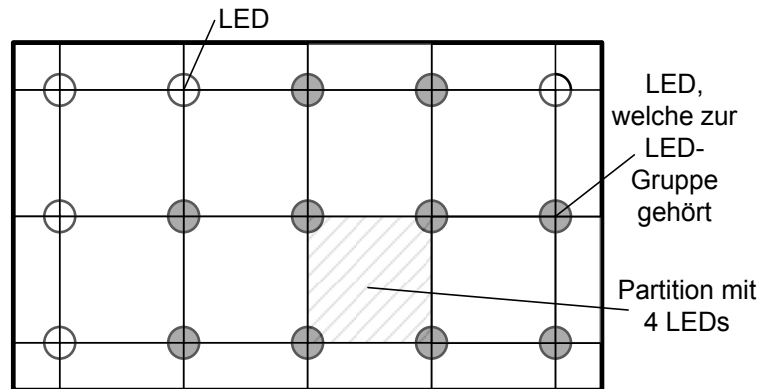


Abbildung 4-5: Partition mit erweiterter LED-Gruppe

Die Betrachtung eines Pixels impliziert die Betrachtung seiner LED-Gruppe und seiner dominierenden LED. Kleine Einflüsse können vernachlässigt werden, so dass die Anzahl der betrachteten LEDs, welche Einfluss auf einen Pixel ausüben, einer Begrenzung unterliegt.

Je größer die vernachlässigten Einflüsse sind, desto höher wird in der Regel der Leistungsverbrauch. Auf der anderen Seite verringert eine Begrenzung der beeinflussenden LEDs die Komplexität der Berechnungen, was sich in einer kleineren Chipfläche und in einer geringeren Laufzeit widerspiegelt.

Entscheidend ist, dass eine LED-Gruppe aus einer begrenzten Anzahl von LEDs besteht, deren Helligkeit variiert wird, so dass die Komplexität der gegenseitigen Abhängigkeit (Interdependenz) der LEDs drastisch reduziert werden kann.

4.3 Bedingung für einen Approximationsalgorithmus

Die in Formel 3.3 eingeführte Einflussmatrix A , welche den Einfluss jeder LED auf jeden Pixel beschreibt, soll nun etwas genauer untersucht werden. Abbildung 4-6 zeigt die grafische Beschreibung der LSFs verschiedener LEDs bei einem Direct-Lit Backlight (oben) und die daraus resultierende Einflussmatrix (unten).

Man erkennt, dass jede LED beim Direct-Lit Backlight nur einen kleinen Teil des Displays merklich beeinflusst. Dadurch sind eine Vielzahl der Einträge der Einflussmatrix A null bzw. sehr klein. Man spricht in diesem Fall von einer dünnbesetzten Matrix.

4.3 Bedingung für einen Approximationsalgorithmus

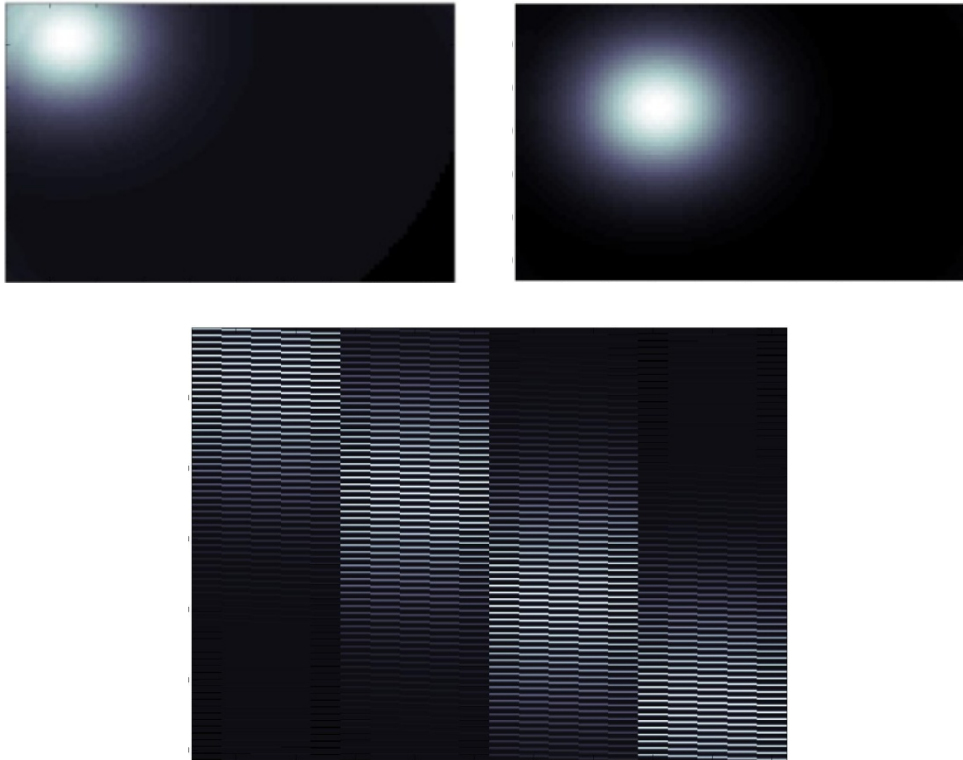


Abbildung 4-6: LSFs verschiedener LEDs bei einem Direct-Lit Backlight (oben) und die daraus resultierende Einflussmatrix (unten)

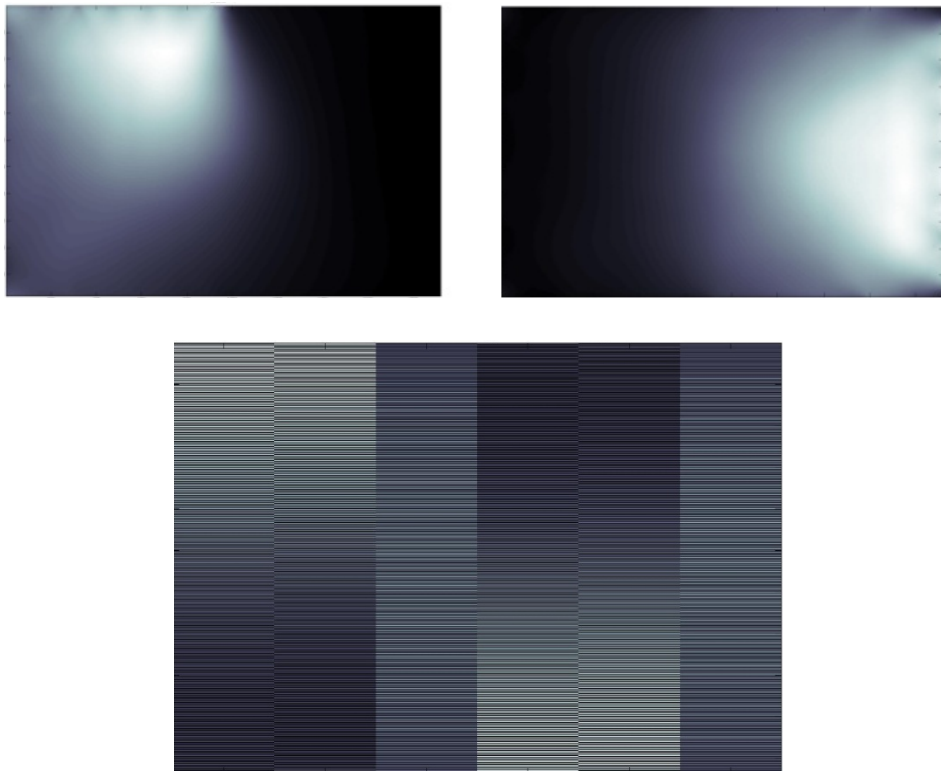


Abbildung 4-7: LSFs verschiedener LEDs bei einem Edge-Lit Backlight (oben) und die daraus resultierende Einflussmatrix (unten)

Die Einflussmatrix bei einem Edge-Lit Backlight unterscheidet sich in zwei Punkten von der oben gezeigten Einflussmatrix eines Direct-Lit Backlights. Die Anzahl der LEDs ist im Allgemeinen um ein Vielfaches kleiner, und die Anzahl der Pixel, die von einer LED beeinflusst werden, ist im Allgemeinen um ein Vielfaches größer.

Abbildung 4-7 zeigt die LSFs verschiedener LEDs bei einem Edge-Lit Backlight (oben) und die daraus resultierende Einflussmatrix (unten). Beim Betrachten der Einflussmatrix fällt auf, dass diese immer noch aus sehr vielen Nullstellen (bzw. sehr kleinen Werten) besteht. Auch hier gibt es also sehr viele Pixel, die im Wesentlichen nur von einer bzw. von einigen wenigen LEDs beeinflusst werden. Es kann also auch im Fall der Einflussmatrix des Edge-Lit Backlights von einer dünnbesetzten Matrix gesprochen werden.

Die in diesem Abschnitt gemachte Beobachtung – die Einflussmatrix A ist dünnbesetzt – erlaubt die Anwendung eines Approximationsalgorithmus [Co 02]. Das bedeutet, es besteht die Möglichkeit, dass eine näherungsweise Lösung des in Formel 3.6 beschriebenen Problems nahe an der optimalen Lösung liegen kann.

4.4 Kurzüberblick zu SSC

Nachdem die Bedingung für einen Approximationsalgorithmus erfüllt ist und mit Abschnitt 4.2 die begrifflichen Grundlagen geschaffen sind, folgt nun eine kurze, einführende Beschreibung der entwickelten Methode zur Berechnung von Local Dimming Backlight.

Um eine möglichst optimale Lösung zu erzielen, sind Iterationen notwendig. Dabei muss die Anzahl der Iterationen jedoch begrenzt und klein sein, damit auch der benötigte HW-Aufwand begrenzt und klein ist.

In Abbildung 4-8 sind die verschiedenen Phasen von SSC zu sehen.

Der Algorithmus setzt sich aus mindestens zwei der in Abbildung 4-8 dargestellten Phasen zusammen, den Lower Bounds (LB) und den Final Duty Cycles (FDC). Die Zwischenphase (Intermediate Phase - IP) ist optional und dient zur Verbesserung der Ergebnisse. Hier werden die LED-Werte in mehreren Iterationen sukzessive an die geforderte Helligkeit angepasst. Die IP ist vor allem dann sinnvoll, wenn viele LEDs jeweils einen signifikanten Einfluss auf viele Pixel haben. Bei Verwendung der IP kann FDC auch als letzte Iteration betrachtet werden.

Im Folgenden werden die drei oben gezeigten Phasen im Detail erläutert. Dabei werden zuerst die LB und dann die FDC vorgestellt, bevor näher auf die IP eingegangen wird.

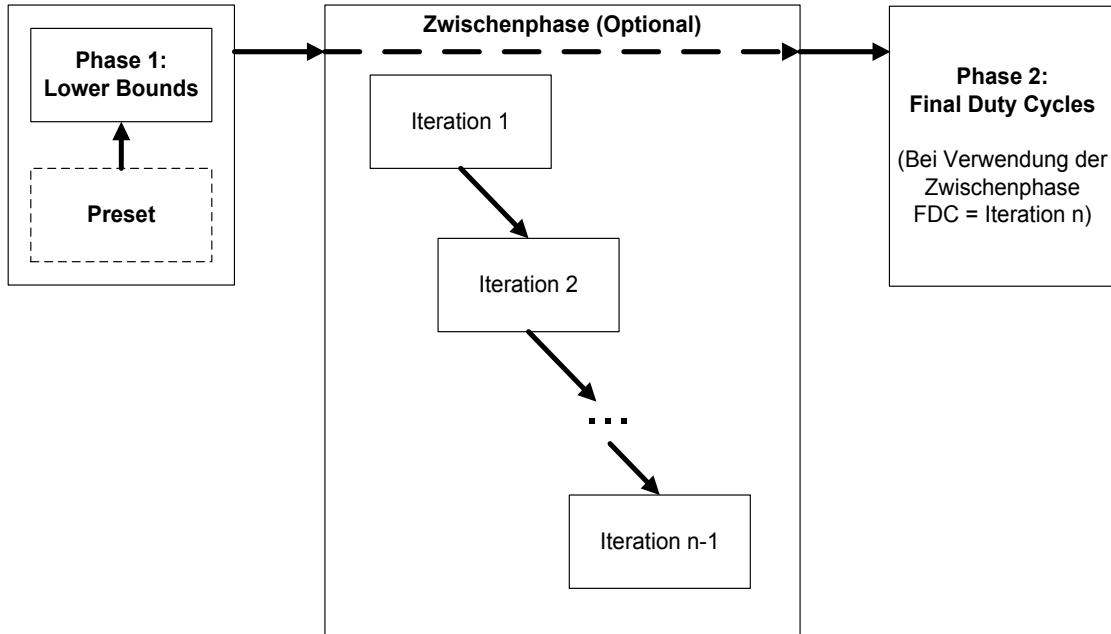


Abbildung 4-8: verschiedene Phasen von SSC

4.5 Phase 1: Preset und Lower Bounds

4.5.1 Einleitung

Ziel der ersten Phase ist es, eine bildabhängige, untere Schranke für die LED-Werte zu bestimmen. Die erste Phase setzt sich aus der einmaligen, bildunabhängigen Ermittlung des Presets und den bildabhängigen LB zusammen.

4.5.2 Optimal Global Dimming und Preset

Zur einmaligen Berechnung des Presets muss zunächst der Begriff des Optimal Global Dimming (OGD) definiert werden. Unter OGD versteht man die optimale Lösung für ein weißes Bild. In den meisten Fällen entspricht das Preset den PWM-Werten für ein ungedimmtes Backlight. Ist dies nicht der Fall, so muss das Ungleichungssystem aus Formel 4.1 mit Hilfe von CPLEX gelöst werden.

$$\begin{bmatrix} a_{1,1} & a_{1,1} & \dots & a_{1,L} \\ a_{2,1} & a_{2,2} & & \\ \dots & & \dots & \\ a_{P,1} & & & a_{P,L} \end{bmatrix} * \begin{bmatrix} x_1 \\ x_2 \\ \dots \\ x_L \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \dots \\ b_P \end{bmatrix} \geq \begin{bmatrix} i_1^{\max} \\ i_2^{\max} \\ \dots \\ i_P^{\max} \end{bmatrix} \quad 4.1$$

Beim darzustellenden Bild i handelt es sich um ein weißes Bild. Jeder Pixel p leuchtet mit seiner maximalen Helligkeit i_p^{\max} .

Der Lösungsvektor x_{pre} , welcher die optimale Lösung der PWM-Werte für dieses Bild darstellt, wird Preset genannt und zur Berechnung der im nächsten Abschnitt eingeführten Lower Bounds benötigt.

4.5.3 Lower Bounds

Wie bereits weiter oben gezeigt, wird ein Pixel immer nur von einer begrenzten Anzahl von LEDs beeinflusst. In den Ecken eines Direct-Lit Displays kann es sogar vorkommen, dass die sich dort befindenden Pixel nur von einer einzelnen LED beeinflusst werden. In diesem Fall wird der LED-Wert der betrachteten LED k mit Hilfe von Formel 4.2 berechnet:

$$x(k) \geq \frac{i_p}{a_p(k)} \quad 4.2$$

Diese besagt, dass der PWM-Wert der LED mindestens so groß sein muss wie der Quotient aus dem geforderten Helligkeitswert des Pixels i_p und dem Einfluss $a_p(k)$ der LED auf diesen Pixel.

Wenn ein Pixel von mehr als einer LED abhängig ist, muss die erzeugte Helligkeit der anderen LEDs ($\ell \neq k$) berücksichtigt werden, welche sich durch das Aufsummieren der einzelnen Helligkeitsbeträge ergibt.

$$\sum_{\ell \neq k} a_p(\ell) \cdot x_{pre}(\ell) \quad 4.3$$

Diese wird dann von der geforderten Helligkeit im Pixel subtrahiert, um den LED-Wert analog zu Formel 4.2 zu ermitteln:

$$x(k) \geq \frac{i_p - \sum_{\ell \neq k} a_p(\ell) \cdot x_{pre}(\ell)}{a_p(k)} \quad 4.4$$

Da die durch die Preset-Werte erzeugte Helligkeit aus Formel 4.3 in der Regel in fast allen Pixeln des Displays der maximalen Helligkeit entspricht, muss zur Berechnung der LB jeweils nur der Bereich von Pixeln in der Nähe der aktuell betrachteten LED k berücksichtigt werden. Die schraffierte Fläche in Abbildung 4-9 zeigt ein Beispiel für einen solchen Bereich.

Für alle Pixel in diesem Bereich bildet man, analog zu Formel 4.4, die Differenz aus dem Sollhelligkeitswert des Pixels und dem durch das Preset erreichten Helligkeitswert im Pixel. Ist diese Differenz positiv, bedeutet dies, dass die k -te LED benötigt wird, um den Sollhelligkeitswert in diesem Pixel zu erreichen. Der einzustellende PWM-Wert wird dann aus dem Quotienten der Differenz und dem Einfluss von $x(k)$ im betrachteten Pixel berechnet. Am Ende der Berechnung von $x(k)$ wird der für diese LED maximal benötigte Wert gespeichert.

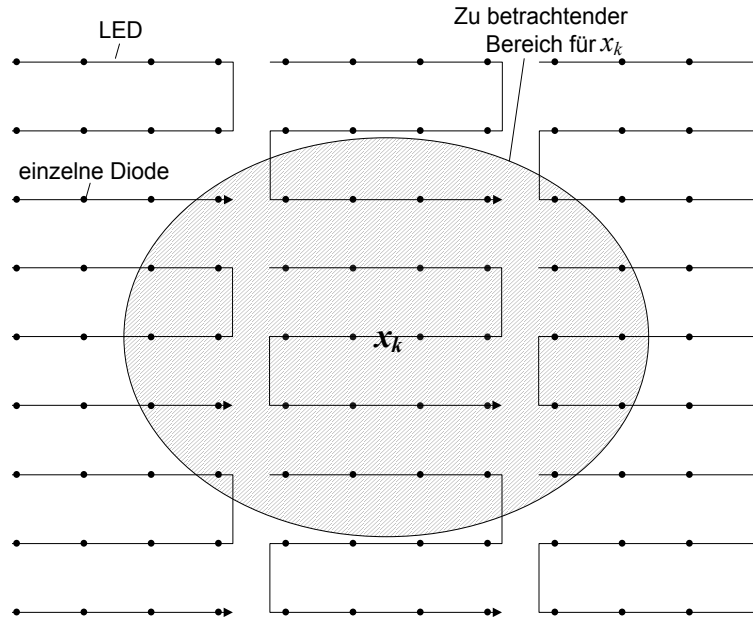


Abbildung 4-9: Grafische Erklärung zur Berechnung der LB

Die beschriebene Prozedur muss einmal für jede LED durchgeführt werden. Die meisten Pixel müssen dabei nur jeweils einmal prozessiert werden.

Da die Berechnung der LB für jede LED unabhängig von den anderen LEDs erfolgt, können diese auch parallel prozessiert werden.

Als Ergebnis der LB erhält man eine vom darzustellenden Bild abhängige, mathematische untere Schranke der LED-Werte. Verglichen mit einer optimalen Lösung bedeutet dies, dass keiner dieser LED-Werte größer als der entsprechende LED-Wert der optimalen Lösung ist. Somit bilden die LB eine ideale Ausgangslage für die in den folgenden Abschnitten beschriebenen nachfolgenden Phasen von SSC.

4.6 Phase 2: Final Duty-Cycles

In der letzten Phase des Algorithmus wird jeder Pixel des Bildes bezüglich seiner Sollhelligkeit überprüft. Hierzu wird die Helligkeit der anderen LEDs in diesem Pixel berechnet.

$$\sum_{\ell \neq k} a_p(\ell) \cdot x(\ell) \quad 4.5$$

Diese wird dann von der geforderten Helligkeit subtrahiert. Falls ein Defizit vorliegt, wird der PWM-Wert der den Pixel dominierenden LED $x(k)$ soweit erhöht, dass die folgende Bedingung erfüllt ist:

$$x(k) \geq \frac{i_p - \sum_{\ell \neq k} a_p(\ell) \cdot x(\ell)}{a_p(k)} \quad 4.6$$

Für den Fall, dass LED $x(k)$ bereits auf den maximalen Wert eingestellt ist, wird der Wert der LED mit dem nächstgrößeren Einfluss auf den Pixel erhöht.

Als Ergebnis der FDC erhält man eine Lösung frei von Clipping. Das bedeutet, dass die in Formel 3.4 beschriebene Forderung ($b \geq i$) erfüllt ist. Im Gegensatz zu den LB besteht in den FDC eine Abhängigkeit zwischen den einzelnen LEDs. Eine Erhöhung eines LED-Wertes bewirkt einen Helligkeitszuwachs in den benachbarten Pixeln und hat somit Auswirkungen auf die Festlegung der anderen LED-Werte. Aus diesem Grund ist es notwendig, eine bestimmte Reihenfolge in der Abarbeitung der Pixel einzuhalten.

Im Folgenden werden zwei Möglichkeiten zur Berechnung der FDC, ein lokaler und ein globaler Optimierungsansatz, präsentiert.

4.6.1 Lokaler Optimierungsansatz

4.6.1.1 Prozessierung einer Partition

Bei Backlights mit einer großen Anzahl von LEDs kann es, wie in Abschnitt 4.2.4 gezeigt, zur Komplexitätsreduktion notwendig sein, das Display in Partitionen zu unterteilen. Diese werden dann nach einem festgelegten Schema prozessiert. Abbildung 4-10 zeigt verschiedene mögliche Abarbeitungsreihenfolgen für eine Partition.

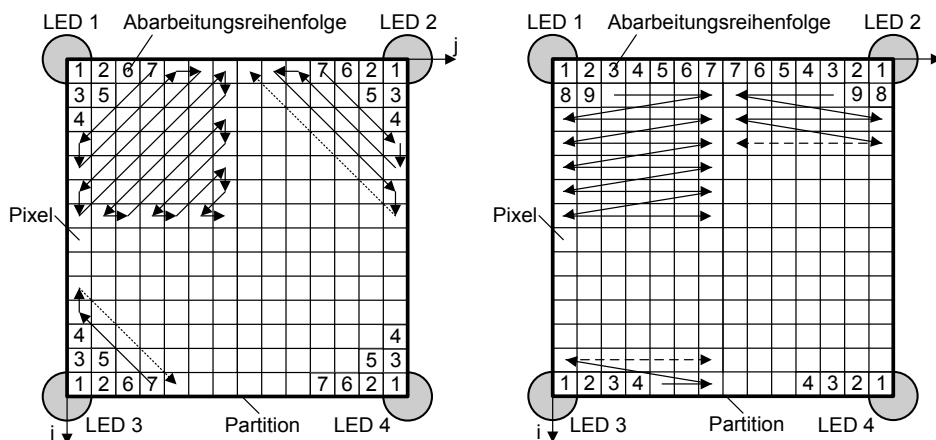


Abbildung 4-10: Verschiedene mögliche Abarbeitungsreihenfolgen einer Partition

Es wird die folgende Reihenfolge empfohlen: Pixel 1 der ersten LED, Pixel 1 der zweiten LED usw. Dann Pixel 2 der ersten LED usw. Am Ende müssen alle Pixel der Partition einmal betrachtet worden sein.

Das oben beschriebene Vorgehen wird dann bevorzugt, wenn eine LED nur in einem kleinen Bereich von Pixeln merklichen Einfluss hat. Ist dies nicht der Fall, so empfiehlt es sich, die Helligkeit, die von den umliegenden LEDs in den Pixeln der Partition (wie in Abbildung 4-5 beschrieben) erzeugt wird, zu berücksichtigen (siehe Abbildung 4-11). Dadurch erhöht sich der

Berechnungsaufwand, allerdings wird durch die Berücksichtigung der umliegenden LEDs mehr Leistung eingespart.

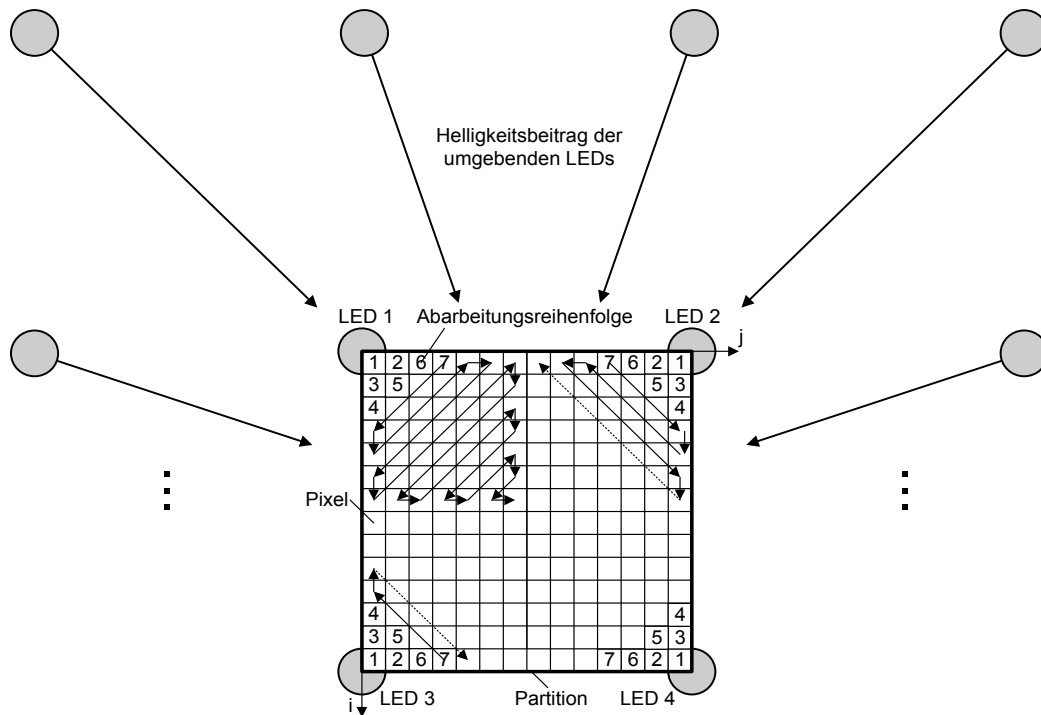


Abbildung 4-11: Helligkeitsbeitrag der umgebenden LEDs in einer Partition

Generell erfolgt die Reihenfolge der Abarbeitung von Pixeln nahe einer LED zu weiter entfernten Pixeln, in anderen Worten also von Pixeln, auf welche die LED einen hohen Einfluss hat, zu Pixeln, die von der LED weniger stark beeinflusst werden. Diese Vorgehensweise basiert auf der bekannten Greedy-Methode [Co 02], welche sich dadurch auszeichnet, dass zu jedem Zeitpunkt der Prozessierung eine für den Moment als optimal erscheinende Entscheidung getroffen wird. Die Konsequenzen auf die Güte des Algorithmus werden in Abschnitt 4.11 diskutiert.

4.6.1.2 Prozessierungsreihenfolge der Partitionen

Ebenso wichtig wie das Einhalten einer gewissen Reihenfolge bei der Abarbeitung der Pixel einer Partition ist die Einhaltung einer gewissen Prozessierungssequenz der Partitionen für die Berechnung der FDC.

Abbildung 4-12 zeigt zwei mögliche Sequenzen. Im oberen Teil der Abbildung werden die Partitionen des Displays in Form eines Schneckenbaus abgearbeitet. Im unteren Teil werden die Partitionen ebenfalls von außen nach innen abgearbeitet, jedoch erfolgt hier ein Sprung nach jeweils der Hälfte einer zurückgelegten Strecke.

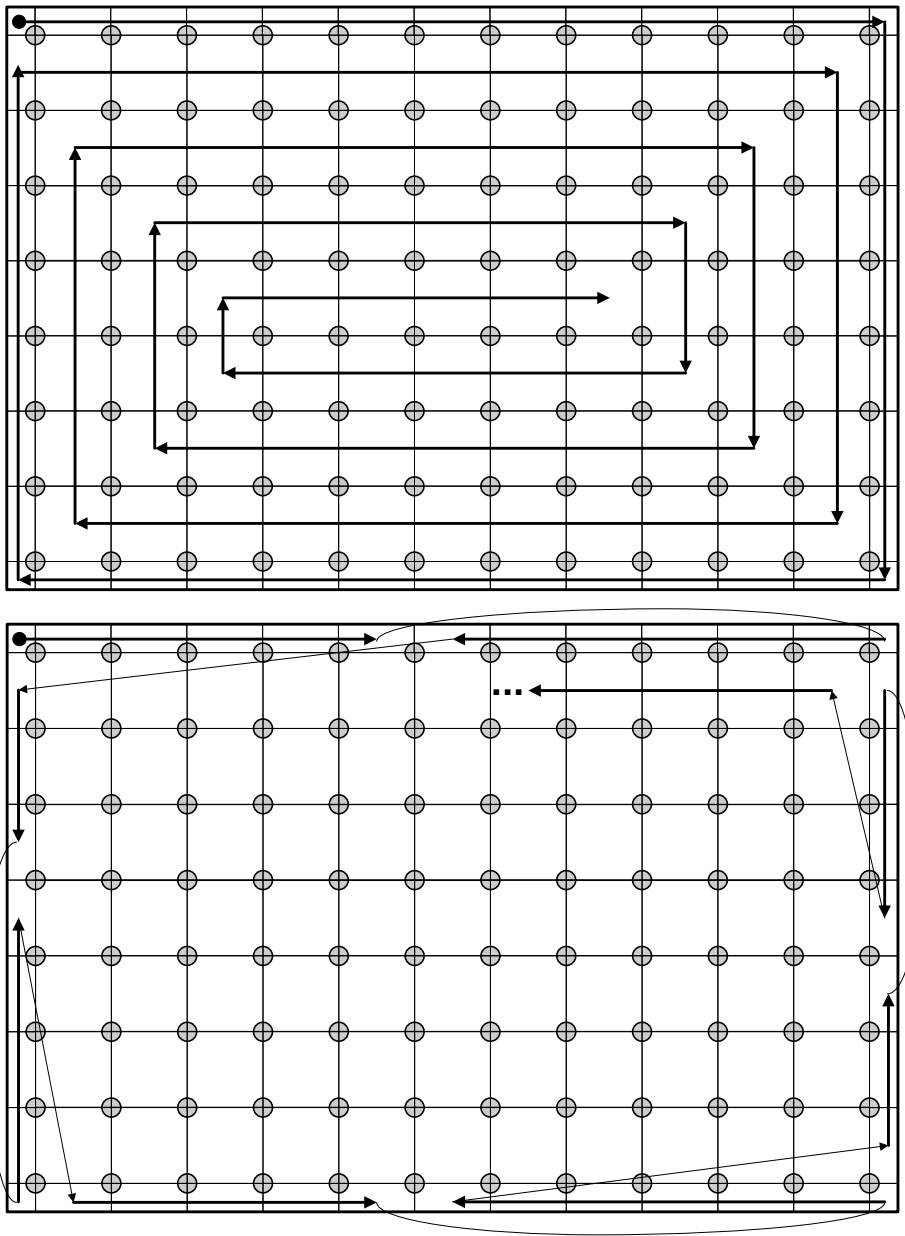


Abbildung 4-12: Verschiedene Prozessierungssequenzen für die Berechnung der LED-Gruppen

Generell gilt, dass die Richtung der Prozessierung von den Ecken bzw. Kanten des Displays hin zur Displaymitte erfolgt. Grund hierfür ist die geringere Interdependenz der LEDs im Randbereich. Deren Werte liegen zu Beginn der FDC näher an ihrer endgültigen Zuweisung als die Werte der LEDs, die sich in der Mitte des Displays befinden.

4.6.2 Globaler Optimierungsansatz

Der in Abschnitt 4.6.1 präsentierte lokale Optimierungsansatz eignet sich dann, wenn die LEDs jeweils nur einen geringen Teil des Displays beeinflussen. Dies trifft im Allgemeinen für Direct-Lit Backlights zu. Werden größere Teile des Displays von einer LED beeinflusst, so

führt die oben beschriebene lokale Optimierung nicht immer zu Ergebnissen nahe an der optimalen Lösung.

Abhilfe schafft ein globaler Optimierungsansatz. Bei diesem wird der Einfluss (fast) aller LEDs bei der Berechnung der aktuellen LED berücksichtigt.

Abbildung 4-13 zeigt die Reihenfolge der Prozessierung eines Edge-Lit Backlights mit sechs LEDs. Zur besseren Übersicht wird das Display in sechs Sektoren unterteilt, jeder Pixel wird also nur von einer LED dominiert.

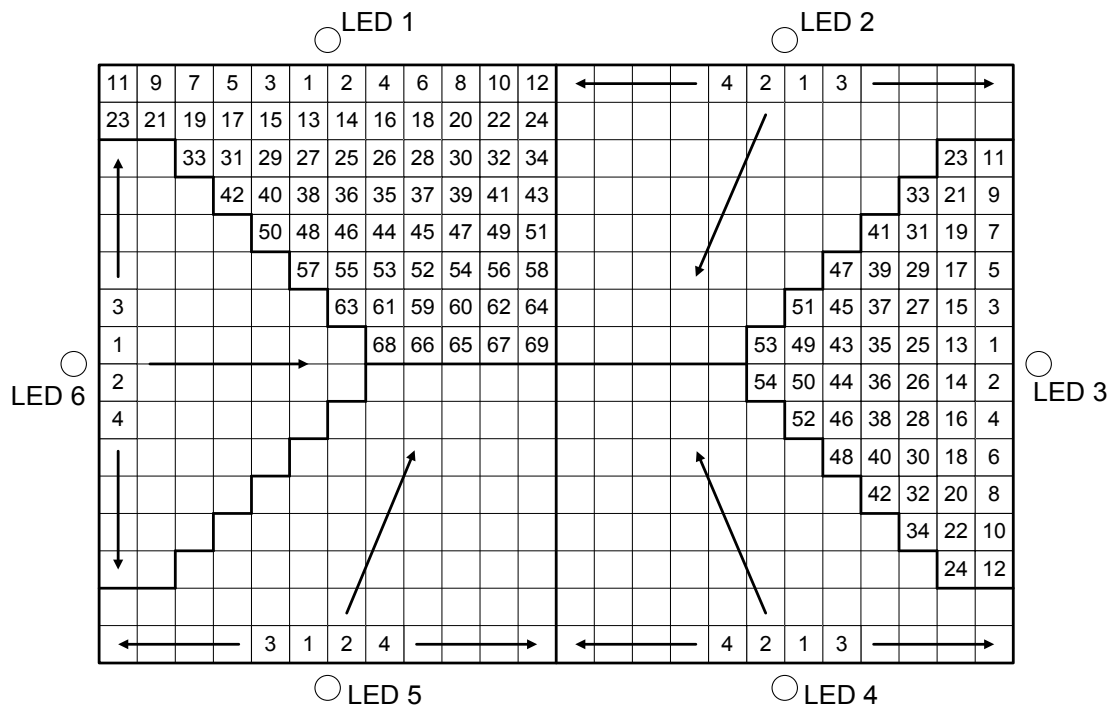


Abbildung 4-13: Reihenfolge der Prozessierung beim globalen Optimierungsansatz am Beispiel eines Edge-Lit Backlights mit 6 LEDs

Manprozessiert reihum zuerst die Pixel, auf welche die dominierende LED den größten Einfluss hat. Zuerst wird Pixel 1 der LED 1 betrachtet, dann Pixel 1 der LED 2 usw. Danach wird Pixel 2 der LED 1 betrachtet usw. Im obigen Beispiel enden die FDC folglich, wenn alle Pixel des Displays einmal betrachtet worden sind, mit Pixel 69 der LED 5, da LED 3 und LED 6 weniger Pixel dominieren.

Eine derartige Vorgehensweise hat zwei Vorteile. Der erste Vorteil ist, dass durch die Berücksichtigung aller LEDs weniger Helligkeit verschwendet wird und somit mehr Leistung eingespart werden kann.

Der zweite Vorteil besteht darin, dass die LEDs reihum erhöht werden. Dadurch nutzt man schon bei der Prozessierung des zweiten Pixels der ersten LED die Helligkeit der anderen LEDs, wodurch weitere Leistung eingespart werden kann.

Auch hierprozessiert man, wie in Abschnitt 4.6.1 dargestellt, von Pixeln nahe der dominierenden LED zu weiter entfernten Pixeln.

4.6.3 Parallele Prozessierung der FDC

Bei großen Displayauflösungen und Video kann zur Beschleunigung der Berechnung eine parallele Prozessierung notwendig sein. Dies stellt bei den LB kein Problem dar, weil die LED-Werte unabhängig voneinander bestimmt werden. Anders verhält es sich bei den FDC. Hier liegt eine Interdependenz zwischen den LEDs vor.

Betrachtet man den unteren Teil der Abbildung 4-12 genauer, fällt auf, dass die räumlich nicht zusammenhängenden Partitionen unabhängig voneinander berechnet werden. Daher ist es möglich, diese Partitionen parallel zu prozessieren.

Eine Möglichkeit zur parallelen Berechnung wird nun präsentiert. Das Display wird dazu in verschiedene Bereiche unterteilt, welche parallel zueinander berechnet werden. Abbildung 4-14 zeigt eine parallele Prozessierung zweiten Grades.

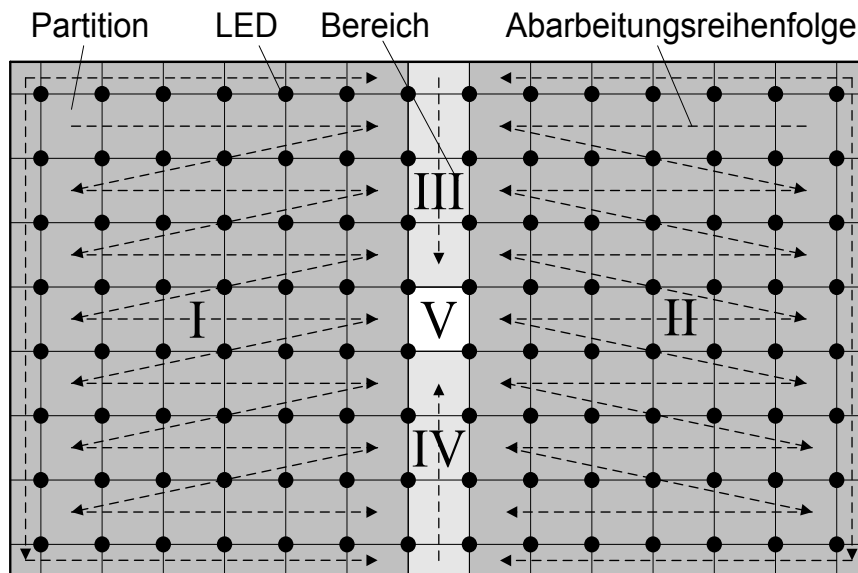


Abbildung 4-14: Parallele Prozessierung zweiten Grades

Bei der parallelen Prozessierung zweiten Grades werden zuerst die Bereiche I und II parallel abgearbeitet. Danach werden parallel die Bereiche III und IV behandelt, bevor schließlich Bereich V berechnet wird.

Die Berechnung bei der parallelen Prozessierung vierten Grades (siehe Abbildung 4-15) verläuft analog. Zuerst werden die Bereiche I, II, III und IV parallel, dann werden die Bereiche V, VI, VII und VIII parallel und schließlich Bereich IX berechnet. Auch hier gilt, sowohl bei der Abarbeitung der Pixel in den Bereichen als auch bei der Abfolge der Bereiche, dass die Prozessierung immer von den Ecken bzw. Rändern des Displays zur Mitte hin erfolgt. Durch eine parallele Prozessierung 4. Grades erhöht sich der Hardwareaufwand um das Vierfache. Jedoch wird die Berechnungszeit auf ein Viertel reduziert, wodurch die Framerate um das Vierfache erhöht wird. Zwischen der Güte der Ergebnisse der parallelen und der sequentiellen Abarbeitung besteht nur ein vernachlässigbarer Unterschied. Die relative Abweichung ist in der Regel kleiner als 0,1%.

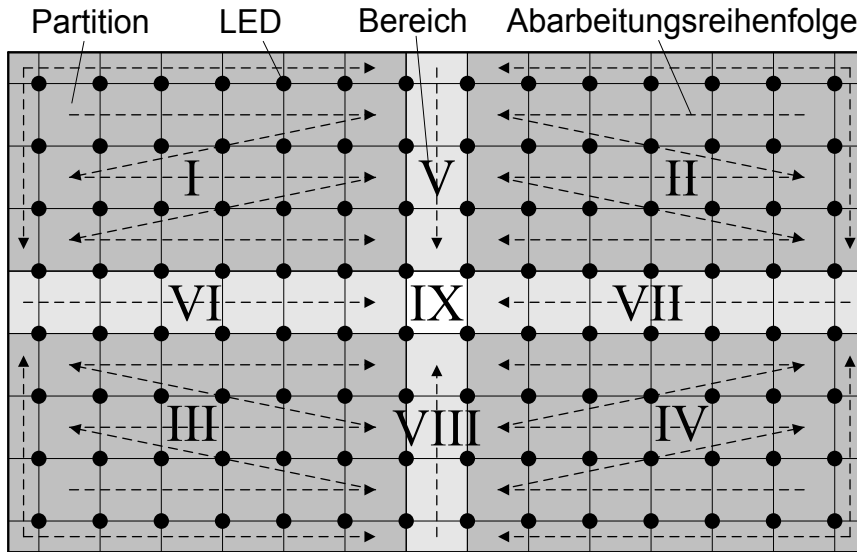


Abbildung 4-15: Parallele Prozessierung vierten Grades

4.7 Zwischenphase

4.7.1 IP mit globalem Optimierungsansatz

Die in Abbildung 4-8 gezeigte IP besteht aus einer festgelegten Anzahl von Iterationen, in welchen die LED-Werte sukzessive an die Bildhelligkeit angepasst werden. Die letzte Iteration n der IP entspricht den im vorigen Abschnitt präsentierten FDC. Die IP ist vor allem dann sinnvoll, wenn viele Werte der Einflussmatrix A aus Formel 3.3 ungleich null sind. Physikalisch gesprochen bedeutet dies, dass die LEDs viele Pixel beeinflussen. Eine Prozessierung der LB und FDC führt dann in der Regel nicht zu zufriedenstellenden Ergebnissen. Der im Folgenden gezeigte Lösungsweg beruht, in modifizierter Form, auf dem im vorigen Abschnitt gezeigten globalen Optimierungsansatz.

Die Abarbeitungsreihenfolge der Pixel ist identisch mit der im globalen Optimierungsansatz, allerdings müssen nicht alle Pixel in jeder Iteration berücksichtigt werden. Der große Unterschied ist, dass nun das Helligkeitsdefizit

$$\Delta_p = i_p - \sum_{\ell=1}^L a_p(\ell) \cdot x(\ell) \quad 4.7$$

nicht mehr in einem Schritt behoben wird, sondern sukzessive. Hierzu wird der Sensibilitätsfaktor $\lambda \in (0,1)$ eingeführt, der mit der zu erreichenden Helligkeit i_p multipliziert wird. Durch Anwendung von Formel 4.8 wird der neue Wert von $x(k)$ festgelegt.

$$\Delta x(k) = \frac{\lambda \cdot i_p - \sum_{\ell=1}^L a_p(\ell) \cdot x(\ell)}{a_p(k)}, \text{ if } (\Delta x(k) > 0) \rightarrow x(k) = x(k) + \Delta x(k) \quad 4.8$$

Alternativ zu Formel 4.8 kann auch die folgende, weniger aufwändige Berechnung verwendet werden. Statt der Multiplikation bildet man nun eine Differenz aus der zu erreichenden Helligkeit und einem Offsetwert δ .

$$\Delta x(k) = \frac{(i_p - \delta) - \sum_{\ell=1}^L a_p(\ell) \cdot x(\ell)}{a_p(k)}, \text{ if } (\Delta x(k) > 0) \rightarrow x(k) = x(k) + \Delta x(k) \quad 4.9$$

Falls die maximal erlaubte Helligkeit einer LED erreicht ist und dennoch ein Helligkeitsdefizit im Pixel besteht, so wird die LED mit dem zweithöchsten Einfluss im betrachteten Pixel erhöht usw.

Für den Fall, dass zwei oder mehr LEDs nahezu identischen Einfluss auf einen Pixel haben, findet, je nach Vorgehensweise, eine der beiden folgenden Formeln Verwendung.

$$\Delta x(k) = \frac{\lambda \cdot i_p - \sum_{\ell=1}^L a_p(\ell) \cdot x(\ell)}{\sum_{queue} a_p(q)}, \text{ if } (\Delta x(k) > 0) \rightarrow x(k) = x(k) + \Delta x(k) \quad 4.10$$

$$\Delta x(k) = \frac{(i_p - \delta) - \sum_{\ell=1}^L a_p(\ell) \cdot x(\ell)}{\sum_{queue} a_p(q)}, \text{ if } (\Delta x(k) > 0) \rightarrow x(k) = x(k) + \Delta x(k) \quad 4.11$$

Alle den Pixel dominierenden LEDs $x(q)$ werden um $\Delta x(q)$ erhöht.

Selbstverständlich muss in jeder Iteration der Sensibilitätsfaktor λ erhöht werden bzw. der Offsetwert δ erniedrigt werden. Für die letzte (n -te) Iteration muss $\lambda = 1$ bzw. $\delta = 0$ sein.

Die in den Formeln 4.8 und 4.9 beschriebenen Vorgehensweisen führen zu annähernd identischen Ergebnissen. Die zweite Vorgehensweise ist für eine Implementierung in Hardware zu bevorzugen, da eine Subtraktion weniger Zeitaufwand und Chipfläche benötigt als eine Multiplikation.

Nach Beendigung der n -ten Iteration liegen Ergebnisse frei von Clipping und nahe der optimalen Lösung vor.

4.7.2 Festlegung des Startwertes und der Schrittweite von λ bzw. δ

Der Startwert von λ bzw. δ wird durch die Beschaffenheit der Einflussmatrix und das damit verbundene Ergebnis der LB bestimmt, welches in der Regel bereits für eine hohe Helligkeit in den Pixeln sorgt. Je dünner die Einflussmatrix besetzt ist, desto höher kann der Startwert von λ bzw. desto niedriger kann der Startwert von δ gewählt werden. Der optimale Startwert für ein Display wird empirisch ermittelt.

Durch eine nichtlineare Erhöhung von λ bzw. eine nichtlineare Verringerung von δ in den weiteren Iterationsschritten wird erreicht, dass die Resultate bei gleicher Schrittzahl näher an der optimalen Lösung liegen und somit mehr Leistung eingespart bzw. die Anzahl der Iterationen bei gleichbleibender Leistungersparnis reduziert werden kann. Dabei gilt, dass in den ersten Iterationen eine höhere Schrittweite erfolgen darf als in den letzten. Der optimale Verlauf von λ bzw. δ bei einer festgelegten Anzahl von Iterationen wird, genau wie der Startwert, empirisch ermittelt.

Abbildung 4-16 zeigt einen qualitativen Verlauf von λ und δ bei sieben Iterationsschritten. Im letzten Iterationsschritt ist $\lambda = 1$ bzw. $\delta = 0$.

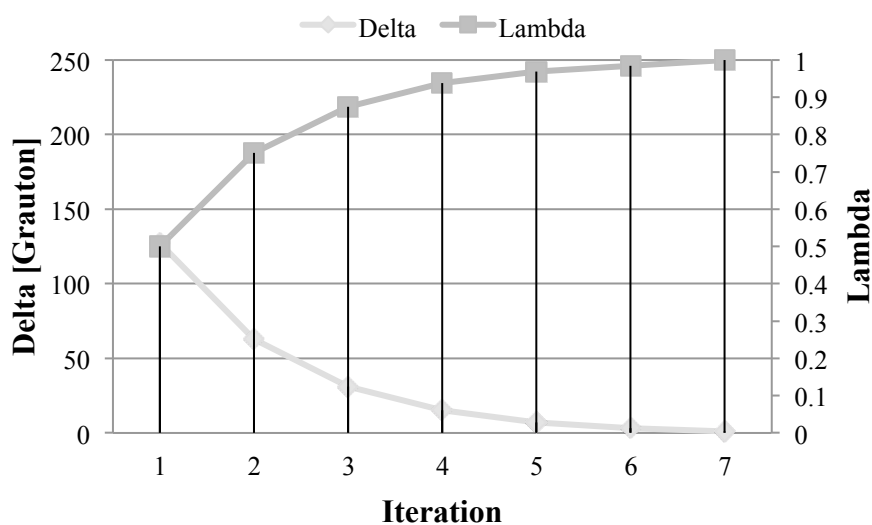


Abbildung 4-16: Verlauf von λ und δ bei sieben Iterationsschritten der IP (der Offsetwert δ wird hier in Grautönen dargestellt)

4.7.3 Zusammenfassung der IP

Um den Algorithmus in deterministischer Zeit abzuarbeiten, muss die IP nach einer festgelegten Anzahl von Iterationen beendet sein. Die letzte Iteration der IP ist äquivalent zur Berechnung der FDC, da hier $\lambda = 1$ bzw. $\delta = 0$ ist. Durch das sukzessive Erhöhen der LED-Werte wird die vom Backlight generierte Helligkeit langsam an die benötigte Helligkeit angenähert. Gerade dann, wenn die LEDs große Bereiche des Displays beeinflussen, führt die IP zu erheblich besseren Ergebnissen als die bloße Verwendung von LB und FDC.

4.8 Unterscheidung von SSC vom Stand der Technik

Der hier vorgestellte Algorithmus unterscheidet sich wesentlich vom Stand der Technik (siehe Abschnitt 2.3). Basiert die dortige Vorgehensweise auf Methoden der Bildverarbeitung, so basiert SSC auf einem physikalischen Modell des Backlights und ist dem Gebiet der diskreten Optimierung zuzuordnen.

Beim Stand der Technik werden die LED-Werte durch eine Mittelwertbildung der zur LED zugeordneten Bereiche festgelegt, bei SSC dagegen werden die LED-Werte analog zur physikalischen Ausbreitungsrichtung des Lichtes berechnet (siehe Abbildung 4-17).

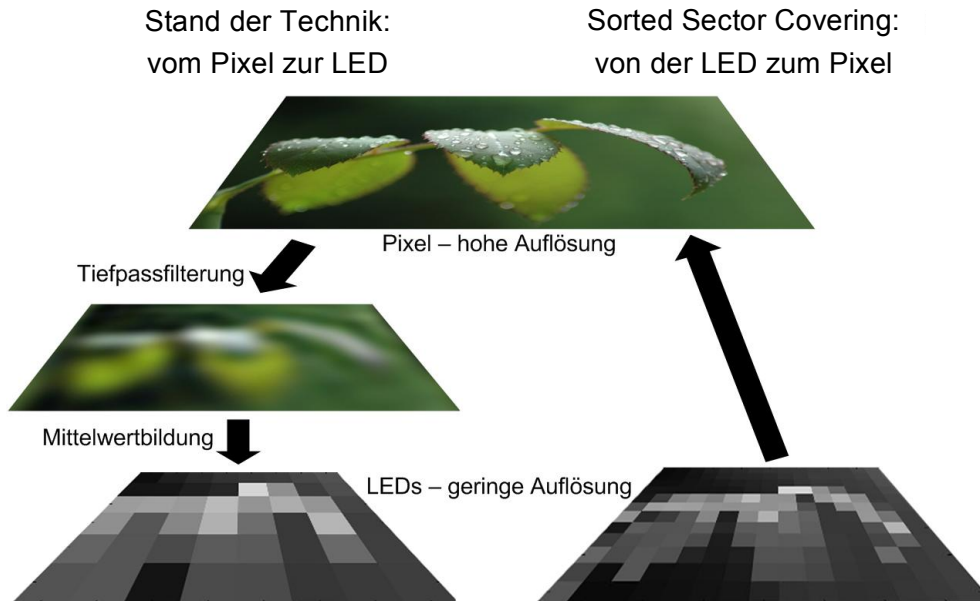


Abbildung 4-17: Vergleich der Prozessierungsrichtung vom Stand der Technik mit der von SSC

Bei SSC wird die aktuell erreichte Helligkeit im Pixel anhand der LSF der LEDs und deren LED-Werten festgelegt. Das hat den Vorteil, dass SSC an jede erdenkliche Backlight-Konstellation adaptierbar ist.

Aufgrund der einfachen Struktur und der Tatsache, dass auch das Übersprechen der LEDs im Pixel während der Berechnung der LED-Werte berücksichtigt wird, ist die SSC-Methode als einzige in der Lage, in Echtzeit Ergebnisse zu berechnen, welche frei von Clipping und gleichzeitig nahe an der optimalen Lösung sind.

4.9 Merkmale von SSC

Der Name des Algorithmus, Sorted Sector Covering, setzt sich aus drei der insgesamt acht wesentlichen Merkmale des Algorithmus zusammen:

1. Der Namensbestandteil „Sorted“ lässt sich dadurch erklären, dass die jeweilige Abarbeitungsreihenfolge der Pixel (und gegebenenfalls der Partitionen) festgelegt ist. Die Reihenfolge richtet sich nach den Einflüssen der LEDs auf die Pixel.
2. Die Pixel des Displays sind verschiedenen Sektoren („Sector“) zugeordnet, um die Komplexität des zu lösenden Problems einzuschränken.

3. Eine Bedingung, die an den Algorithmus gestellt wird, ist, dass die Ergebnisse frei von Clipping sein sollen. Alle Pixel werden mindestens mit der geforderten Helligkeit abgedeckt („Covering“).
4. Das vierte wesentliche Merkmal des Algorithmus ist, dass die Laufzeit von SSC linear abhängig von der Anzahl der Pixel ist (siehe hierzu auch Abschnitt 4.11.2). Der Einfluss der LEDs auf die Laufzeit ist abhängig von der Beschaffenheit der Einflussmatrix, spielt jedoch nur eine untergeordnete Rolle.
5. Bei der Betrachtung eines Pixels wird immer die LED erhöht, welche den größten Einfluss auf den Pixel hat. Falls diese dominierende LED bereits auf den maximalen PWM-Wert eingestellt ist, wird die LED mit dem zweithöchsten Einfluss erhöht usw. Die Vorgehensweise basiert auf der bekannten Greedy-Methode.
6. Bei der Berechnung der LED-Werte wird der Helligkeitsbeitrag der anderen zur LED-Gruppe gehörenden, nicht dominierenden LEDs in dem aktuell betrachteten Pixel mitberücksichtigt. Die Anzahl der LEDs der LED-Gruppe, deren Helligkeit variiert wird, ist begrenzt, so dass die Komplexität der gegenseitigen Abhängigkeit der LEDs drastisch reduziert werden kann.
7. Alle LED-Werte werden im Verlauf des Algorithmus erhöht, niemals verringert. Aus diesem Grund bleiben die einmal abgedeckten Pixel immer abgedeckt.
8. Ziel von Local Dimming LED Backlight ist das Einsparen von Energie. Energie ist eine physikalische Größe. Daher entspricht die Blickrichtung des Algorithmus der physikalischen Ausbreitungsrichtung des Lichtes – von der LED zum Pixel. Somit ist der Algorithmus für das lokale Dimmen aller Arten und Strukturen von Backlights geeignet.

4.10 Einsatzmöglichkeiten von SSC

Abbildung 4-18 zeigt die verschiedenen möglichen Konstellationen auf, bei denen der SSC-Algorithmus eingesetzt werden kann.

In dieser Arbeit werden die hier hervorgehobenen Felder – das lokale Dimmen von weißen LEDs sowohl bei direktem als auch bei indirektem Backlight – behandelt.

0D- bzw. 1D-Dimming stellt eindeutig eine Vereinfachung des 2D-Dimmings dar und kann daher ohne Probleme von SSC angewendet werden. Unter 3D-Dimming versteht man das lokale Dimmen des Backlights mit roten, grünen und blauen LEDs. Die Farbe wird hierbei als zusätzlicher Freiheitsgrad betrachtet. In Abschnitt 10.1 wird explizit auf diese Art des Dimmens eingegangen.

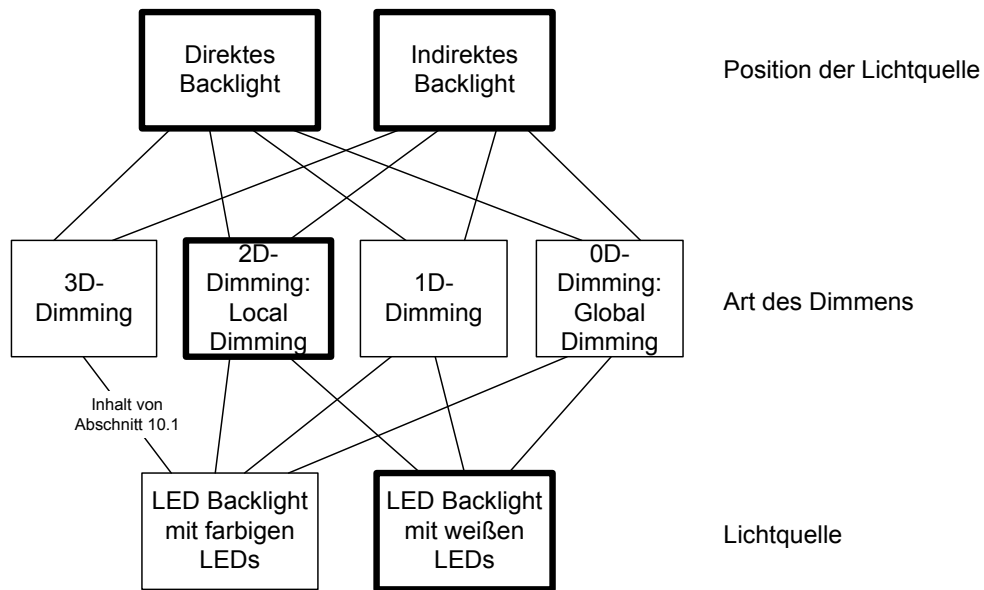


Abbildung 4-18: Einsatzmöglichkeiten von SSC – hervorgehoben die in dieser Arbeit beschriebene Konstellation

4.11 Güte von SSC

Abschließend werden in diesem Abschnitt Aussagen über die Güte (auch Performanz genannt) von SSC getroffen. Unter der Güte eines Approximationsalgorithmus versteht man die Qualität der Ergebnisse (in diesem Fall die Abweichung von der optimalen Lösung). Dabei darf natürlich nicht der Rechenaufwand des Algorithmus (Laufzeit) außer Acht gelassen werden.

4.11.1 Qualität der Ergebnisse

Der in diesem Kapitel beschriebene SSC-Algorithmus verspricht, dass die Ergebnisse nahe an der optimalen Lösung liegen. Optimale Ergebnisse können jedoch durch das Zusammenspiel der folgenden drei Gründe in der Regel nicht garantiert werden:

1. Anlehnung an Greedy-Methode

Der erste Grund ist, dass zu jedem Zeitpunkt, sprich bei jeder Betrachtung eines Pixels, eine in diesem Moment als optimal erscheinende Entscheidung getroffen wird (siehe Merkmal 5 des SSC-Algorithmus). Je geringer der Einfluss der dominierenden LED in einem Pixel ist, desto wahrscheinlicher ist es, dass durch ein Erhöhen dieses LED-Wertes Energie verschwendet wird. Je mehr LEDs also einen signifikanten Einfluss auf einen Pixel ausüben, desto größer ist die zu erwartende Abweichung von der optimalen Lösung.

2. Festgelegte Pixelreihenfolge

Im Allgemeinen verhindert die vorher festgelegte Pixelreihenfolge (siehe Merkmal 1 des SSC-Algorithmus) bei der Prozessierung eine optimale Lösung. Dies wird an einem einfachen Beispiel mit zwei LEDs und acht Pixeln verdeutlicht (siehe Abbildung 4-19).

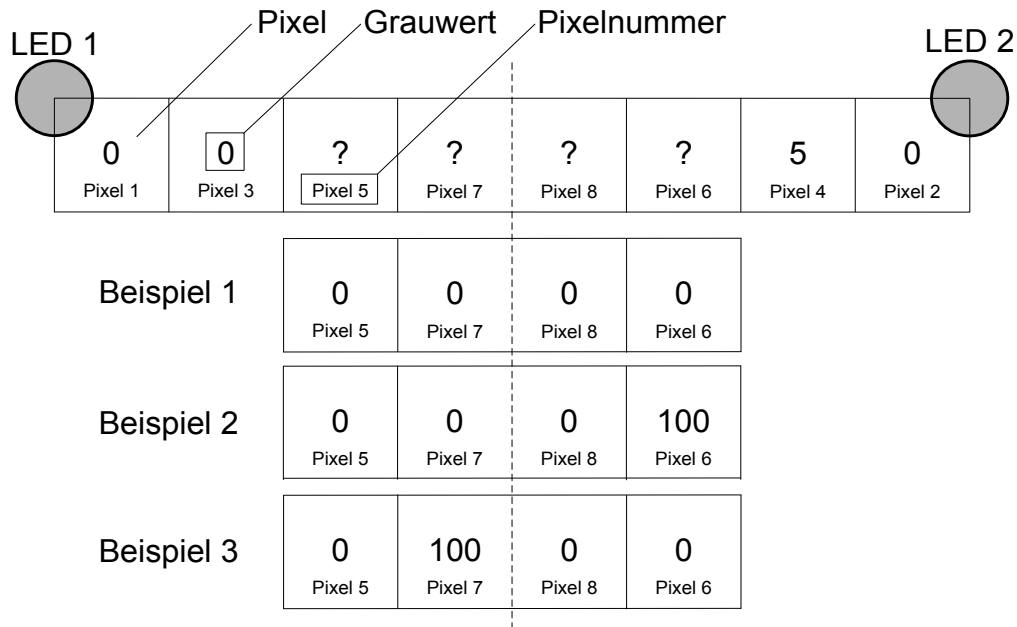


Abbildung 4-19: Beispiele zur festgelegten Pixelreihenfolge

Die ungeraden Pixel werden von LED 1, die geraden Pixel von LED 2 dominiert. Beide LED-Werte sind zu Beginn null. Die Abarbeitungsreihenfolge der Pixel entspricht der Pixelnummer. Ziel ist es, die Verlustleistung zu minimieren, also die Summe der beiden LED-Werte möglichst gering zu halten. Nach den oben gezeigten vier prozessierten Pixeln bleibt der Wert von LED 1 gleich null, da Pixel 1 und Pixel 3 jeweils einen Wert von null haben. LED 2 wird erhöht, da Pixel 4 einen Wert von fünf aufweist. Für die ersten vier Pixel wurde also eine bisher optimale Lösung gefunden. Anhand der drei Beispiele in Abbildung 4-19 wird klar, dass diese bisherige Lösung im weiteren Verlauf nicht unbedingt zu einer optimalen Lösung führt.

In Beispiel 1 haben alle Pixel den Wert null. Die LED-Werte bleiben somit unverändert. Beispiel 2 führt zu keiner Erhöhung bei LED 1, allerdings wird LED 2 erhöht, um den Pixelwert von Pixel 6 abdecken zu können.

Bisher hatte das Erhöhen von LED 2 aufgrund von Pixel 4 keine Auswirkungen auf das endgültige Ergebnis. In Beispiel 3 jedoch wird LED 1 auf einen hohen Wert eingestellt, um Pixel 7 abdecken zu können, während LED 2 nicht verändert werden muss. Die dadurch entstehende Helligkeit von LED 1 in Pixel 4 reicht allerdings aus, um diesen abzudecken. Das Erhöhen von LED 2 führte somit zu einem unnötig hohen Leistungsverbrauch.

Eine weitere entwickelte Möglichkeit zur zusätzlichen Verbesserung der Ergebnisse wird in Appendix C vorgestellt. Dort wird die Abarbeitungsreihenfolge der Pixel abhängig vom Bildinhalt festgelegt. Diese ebenfalls im Rahmen dieser Arbeit entwickelte Methode ist allerdings durch den erhöhten Speicherbedarf und –zugriff nicht für Echtzeitanwendungen geeignet und findet somit bei der Beschreibung und der Implementierung des SSC-Algorithmus keine Berücksichtigung.

3. LED-Werte dürfen nicht verringert werden

Schließlich darf eine einmal getroffene Entscheidung (Erhöhen eines LED-Wertes) nicht wieder revidiert werden. LED-Werte dürfen nur erhöht werden (siehe Merkmal 7 des SSC-Algorithmus).

Die tatsächliche Abweichung zur optimalen Lösung ist abhängig von der Beschaffenheit der Einflussmatrix. Je weniger LEDs einen signifikanten Einfluss auf einen Pixel ausüben, desto besser sind in der Regel die Ergebnisse. Dank der IP wird jedoch die Abweichung von der optimalen Lösung auch bei Backlights mit vielen LEDs, von denen jede wiederum viele Pixel signifikant beeinflusst, beschränkt. Durch das sukzessive Erhöhen der LED-Werte wird verhindert, dass eine LED in einem Schritt zu stark erhöht wird. Dies wird in Abschnitt 8.2 anhand von verschiedenen Prototypen gezeigt.

4.11.2 Laufzeit

Die Laufzeit von SSC ist abhängig von der Anzahl der zu prozessierenden Pixel, der Anzahl der LEDs und der Anzahl der Iterationsschritte, wobei LB und FDC jeweils als ein Iterationsschritt betrachtet werden. Durch die Vorgabe einer zu erreichenden Qualität kann in der Regel die Anzahl der Iterationsschritte in der IP festgelegt werden. Daher handelt es sich bei der Anzahl der LEDs und der Zahl der Iterationsschritte um Konstanten, welche noch dazu um Größenordnungen kleiner sind als die Anzahl der Pixel. Darüber hinaus werden im Allgemeinen nicht alle LEDs bei der Berechnung eines Pixels berücksichtigt. Behandelt man also die Anzahl der Iterationsschritte und die Anzahl der LEDs als Konstanten, ergibt sich daraus, dass die Laufzeit von SSC linear abhängig von der Anzahl der Pixel ist.

4.12 Fazit

Der präsentierte SSC-Algorithmus stellt eine eigenständige Methode zur Berechnung von Local Dimming Backlight dar. Diese unterscheidet sich grundlegend vom Stand der Technik und erzeugt clippingfreie Ergebnisse nahe an der optimalen Lösung. Der Algorithmus ist an alle erdenklichen Backlight-Konstellationen adaptierbar. Durch die einfachen mathematischen Operationen ist eine effiziente Implementierung in Hardware zu erwarten.

Es wurde gezeigt, dass die Laufzeit des Algorithmus proportional zur Anzahl der zu prozessierenden Pixel ist. Aus diesem Grund wird im folgenden Kapitel eine Möglichkeit aufgezeigt, wie diese Anzahl drastisch reduziert werden kann. Dabei wird der SSC-Optimierer, welcher den Kern der Prozessierung darstellt, mit einem eigens entwickelten Präprozessor, dem Condenser, verbunden.

5 Condenser – Präprozessor für SSC

5.1 Einleitung

In Abschnitt 4.9 wurden die verschiedenen Merkmale des SSC-Algorithmus dargestellt. Die Merkmale 3 und 4 werden nun noch einmal genauer betrachtet:

Merkmale 3: „Die Ergebnisse von SSC sind immer frei von Clipping.“

Dies führt zu perfekten visuellen Ergebnissen.

Da jedoch Clipping in Maßen vom menschlichen Auge nicht als störend empfunden wird, kann ein kontrolliertes Zulassen von Clipping zu einer höheren Leistungsparsnis führen, ohne dass die visuelle Qualität darunter leidet.

Merkmale 4: „Die Laufzeit von SSC ist proportional zur Anzahl der zu prozessierenden Pixel.“

Gäbe es also eine Möglichkeit weniger Pixel zu prozessieren, so würde die Laufzeit von SSC verringert werden.

Der eigens für SSC entwickelte Präprozessor, genannt Condenser, welcher ein kontrolliertes Zulassen von Clipping bei gleichzeitiger Reduktion der Laufzeit gewährleistet, stellt den zweiten Schwerpunkt dieser Arbeit dar und ist Gegenstand dieses Kapitels.

Zunächst werden die Grundlagen des Condensers erläutert. Dann wird gezeigt, wie über die Beschaffenheit der LSFs die maximale Zellgröße des Condensers ermittelt wird. Danach werden verschiedene Möglichkeiten zur Kondensierung der Bilddaten vorgestellt. Darunter fallen verschiedene starre Condenser-Modi und der entwickelte automatische Modus. Aufbauend darauf wird eine Erweiterung zum Condenser, der sogenannte Surcondenser, eingeführt. Abgeschlossen wird das Kapitel mit einem Fazit.

5.2 Grundlagen des Condensers

Wie in Abbildung 5-1 zu sehen ist, wandelt der Condenser das Originalbild in ein kondensiertes Bild um, welches dann von SSC prozessiert wird.

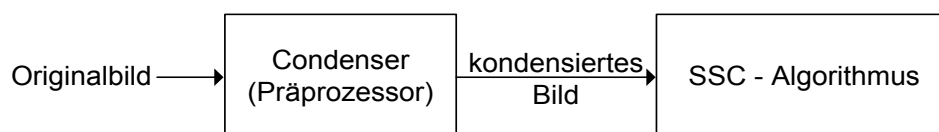


Abbildung 5-1: Blockdiagramm Condenser und SSC

Bei der Kondensierung wird eine rechteckige Zelle von $(s \cdot t)$ Pixeln zu einem kondensierten Pixel $i'_{i',j'}$ zusammengefasst (siehe Abbildung 5-2). Dadurch verringert sich die Anzahl der zu prozessierenden Pixel und somit auch die Laufzeit von SSC.

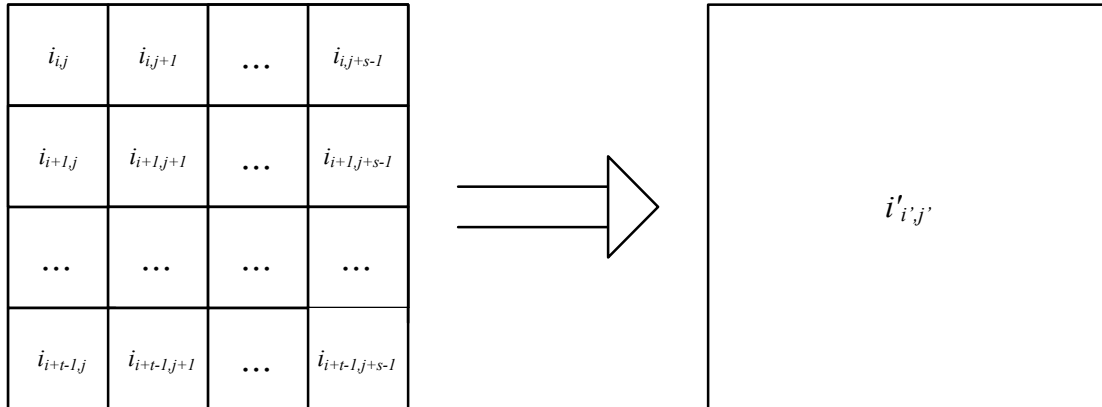


Abbildung 5-2: Kondensierung einer Zelle von $(s \cdot t)$ Pixeln zu einem kondensierten Pixel $i'_{i',j'}$.

Neben den Bilddaten werden natürlich auch $(s \cdot t)$ Einträge der LSFs kondensiert. Abbildung 5-3 zeigt den Zusammenhang zwischen der Kondensierung der LSFs sowie der Bilddaten und deren Auswirkung auf die Laufzeit sowie die Leistungsersparnis.

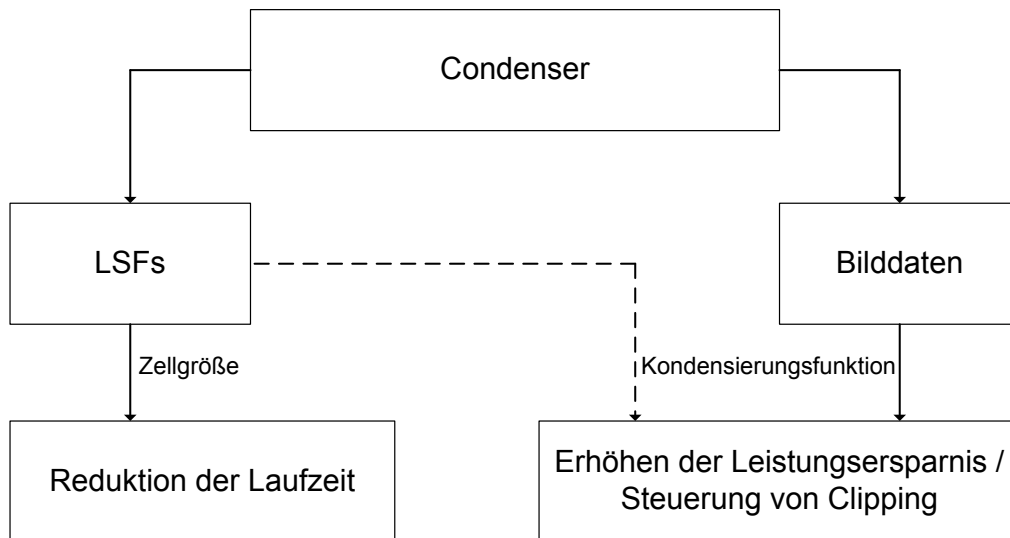


Abbildung 5-3: Zusammenhang zwischen Kondensierung der LSFs sowie der Bilddaten und deren Auswirkung auf die Laufzeit sowie die Leistungsersparnis

Wie in der Abbildung 5-3 zu sehen ist, werden die maximale Zellgröße und die damit verbundene Reduktion der Laufzeit durch die Beschaffenheit der LSFs vorgegeben. Eine Zusammenfassung von beispielsweise $(20 \cdot 20)$ Pixeln zu einem kondensierten Pixel reduziert die Laufzeit auf ein Vierhundertstel.

Die Auflösung des kondensierten Bildes bleibt dennoch um Größenordnungen höher als die LED-Auflösung. Die wesentlichen Bildinhalte bleiben erhalten.

Die Wahl der Funktion, mit der die Bilddaten kondensiert werden, bestimmt die Erhöhung der Leistungsersparnis und steuert das auftretende Clipping.

5.3 Kondensierung der LSFs

5.3.1 Kondensierungsfunktion der LSFs

Neben den bereits in Kapitel 4 beschriebenen Unterschieden der LSFs von Direct-Lit und Edge-Lit Backlights soll nun eine Gemeinsamkeit zwischen den beiden Typen herausgestellt werden. Im oberen Teil von Abbildung 5-4 wird die LSF eines Direct-Lit Backlights (linke Seite) und die eines Edge-Lit Backlights (rechte Seite) gezeigt.

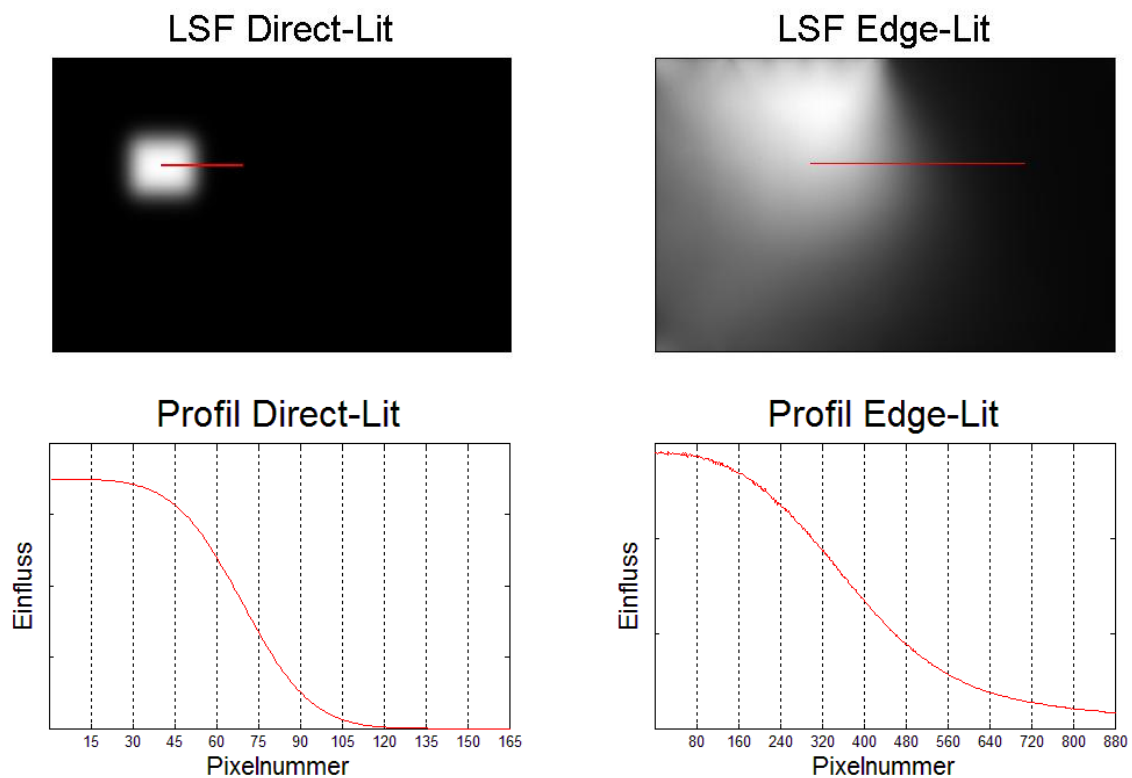


Abbildung 5-4: oben: LSF eines Direct-Lit Backlights (links) und eines Edge-Lit Backlights (rechts); unten: Helligkeitsprofil des markierten Ausschnitts aus dem Direct-Lit Backlight (links) und dem Edge-Lit Backlight (rechts) mit Kennzeichnung der Zellgröße

Betrachtet man jeweils das Helligkeitsprofil eines kleinen Ausschnittes der LSFs (siehe Markierung), so erkennt man, dass der Helligkeitsverlauf bei der Wahl einer geeigneten Zellgröße innerhalb dieser monoton und mit nahezu konstanter Steigung ist (siehe unterer Teil der Abbildung 5-4). Die gewählte Zellgröße beim Direct-Lit Backlight beträgt hier 15 Pixel, die beim Edge-Lit Backlight 80 Pixel.

Möchte man Ergebnisse frei von Clipping erzielen, so müsste der kleinste Wert der LSF innerhalb einer Zelle als kondensierter Wert übernommen werden. Allerdings würde dann auch

Helligkeit nicht berücksichtigt und somit unnötig Leistung verbraucht werden. Je weiter man sich jedoch von diesem Minimum entfernt, desto wahrscheinlicher ist das Auftreten von Clipping.

Im weiteren Verlauf dieser Arbeit wird der arithmetische Mittelwert der Zelle als kondensierter Wert übernommen. Dies führt zwar dazu, dass der Helligkeitsbeitrag einer LED in manchen Pixeln einer Zelle überschätzt bzw. unterschätzt wird, allerdings wird dieses Über- bzw. Unterschätzen durch die Überlagerung mehrerer mit dem arithmetischen Mittelwert kondensierten LSFs in der Regel relativiert. Die Gefahr, dass für den Betrachter sichtbares Clipping entsteht, ist dadurch vernachlässigbar gering.

5.3.2 Bestimmung der maximalen Zellgröße der LSFs

Bei der Bestimmung der maximalen Zellgröße der LSFs gilt das Kriterium, dass sich das Aussehen der originalen und das der kondensierten LSF nicht wesentlich voneinander unterscheiden darf. Die Vorgehensweise wird zunächst am Beispiel eines Direct-Lit Backlights gezeigt (siehe Abbildung 5-5).

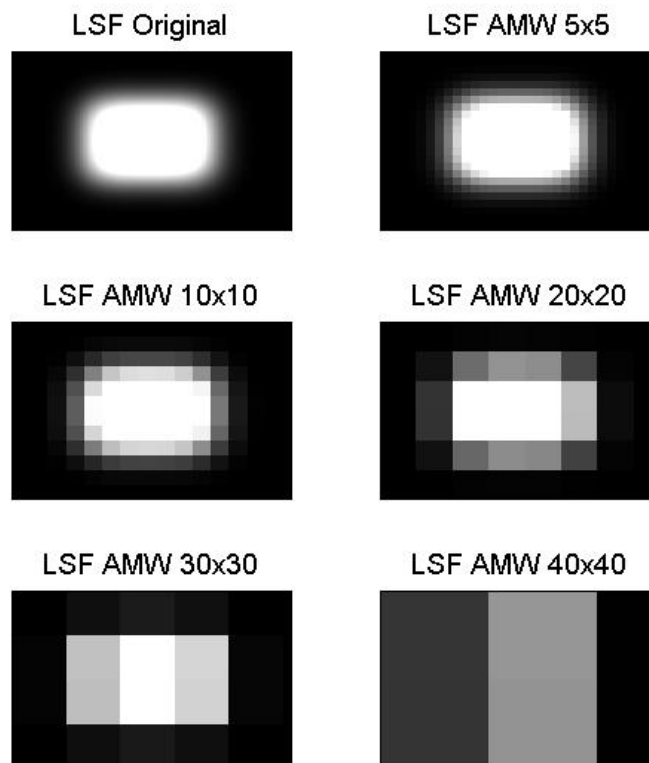


Abbildung 5-5: Vergleich zwischen originaler LSF (oben links) und kondensierten LSFs mit verschiedenen Zellgrößen bei einem Direct-Lit Backlight

Die Zellgröße variiert zwischen (5·5) und (40·40) Pixeln. Vergleicht man die originale LSF mit den kondensierten LSFs, erkennt man ab einer Zellgröße von (20·20) Pixeln einen wesentlichen Unterschied im Aussehen. Daher wäre hier eine maximale Zellgröße zwischen (10·10) und (20·20) Pixeln angemessen.

Nun wird die Vorgehensweise bei der Ermittlung der maximalen Zellgröße am Beispiel einer LSF eines Edge-Lit Backlights wiederholt. Abbildung 5-6 vergleicht die originale LSF mit den kondensierten LSFs bei Variation der Zellgröße zwischen (10·10) und (120·120) Pixeln.

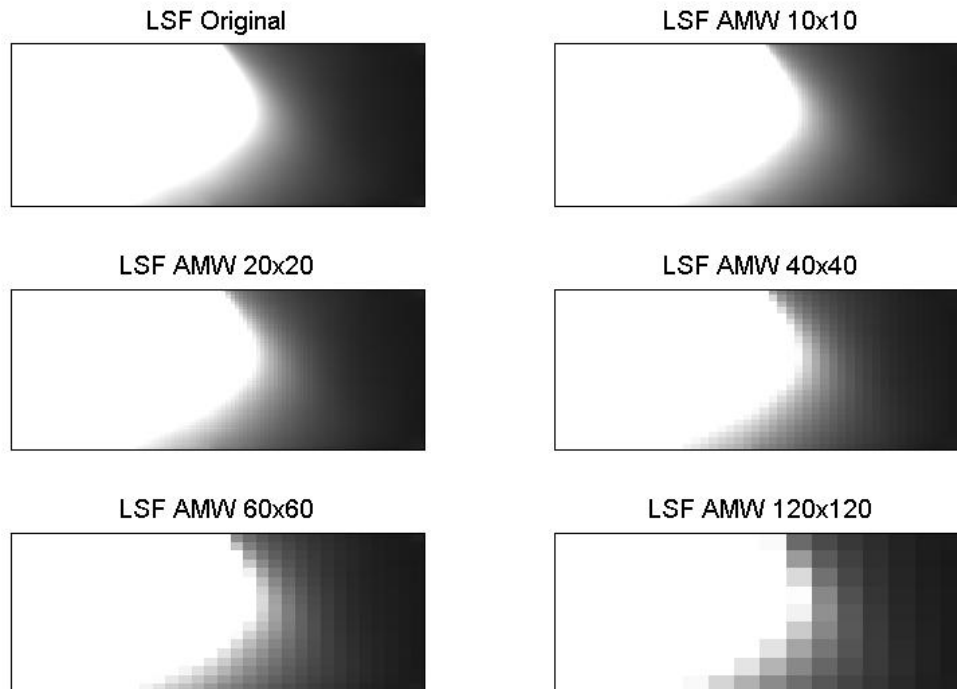


Abbildung 5-6: Vergleich zwischen originaler LSF (oben links) und kondensierten LSFs mit verschiedenen Zellgrößen bei einem Edge-Lit Backlight

Man kommt zu dem Schluss, dass die maximale Zellgröße zwischen (60·60) und (120·120) Pixeln liegen muss.

In diesem Abschnitt wurde gezeigt, dass die maximale Zellgröße abhängig von der Beschaffenheit der LSF ist. Allgemein lässt sich sagen, dass die maximale Zellgröße umso kleiner ist, je größer der Gradient der originalen LSF ist. Daher kann die maximale Zellgröße bei einem Edge-Lit Backlight in der Regel bedeutend größer gewählt werden als bei einem Direct-Lit Backlight.

5.4 Kondensierung der Bilddaten

5.4.1 Starre Condenser-Modi

Im Folgenden werden verschiedene der entwickelten starren Condenser-Modi vorgestellt. Starr bedeutet, dass die Kondensierungsfunktion für alle Zellen des Bildes identisch ist.

5.4.1.1 Clippingfreier Modus

In diesem Modus werden alle Pixel einer Zelle analysiert und der maximale Wert (MAX) aller Pixel schließlich in i'_{ij} abgelegt:

$$i'_{i',j'} = \text{MAX} \begin{pmatrix} i_{i,j} & i_{i,j+1} & \dots & i_{i,j+t-1} \\ i_{i+1,j} & i_{i+1,j+1} & & \\ \dots & & \dots & \\ i_{i+s-1,j} & & & i_{i+s-1,j+t-1} \end{pmatrix} \quad 5.1$$

Durch die Prozessierung auf den jeweils maximalen Wert einer Zelle sind die resultierenden Ergebnisse frei von Clipping⁵ und zeichnen sich durch eine perfekte visuelle Qualität aus.

5.4.1.2 Clippingbehaftete Modi

Um eine höhere Leistungersparnis erzielen zu können, gibt es verschiedene Möglichkeiten, clippingbehaftete Lösungen zu generieren. So kann beispielsweise der kondensierte Pixel i'_{ij} aus dem arithmetischen Mittelwert (AMW) der Zelle gebildet werden:

$$i'_{i',j'} = \text{AMW} \begin{pmatrix} i_{i,j} & i_{i,j+1} & \dots & i_{i,j+t-1} \\ i_{i+1,j} & i_{i+1,j+1} & & \\ \dots & & \dots & \\ i_{i+s-1,j} & & & i_{i+s-1,j+t-1} \end{pmatrix} \quad 5.2$$

Ebenso gut könnte der statistische Mittelwert der Zelle gebildet werden. Aufgrund des geringeren HW-Aufwandes wird in dieser Arbeit jedoch ausschließlich der AMW verwendet.

Eine andere Möglichkeit der Kondensierung wäre, das Mittel aus AMW und MAX zu wählen. Die Kondensierungsfunktion würde dann wie folgt lauten:

$$i'_{i',j'} = \frac{\text{MAX} + \text{AMW}}{2} \quad 5.3$$

Es ist offensichtlich, dass diese Art der Kondensierung weniger Leistung einspart als die reine Mittelwertbildung, dafür aber auch weniger Clipping produziert.

Die drei hier präsentierten starren Condenser-Modi (MAX, AMW und (MAX+AMW)/2) werden in Abbildung 5-7 anhand eines Beispielbildes miteinander verglichen.

Für die drei Modi werden jeweils (20·20) Pixel zu einem kondensierten Pixel zusammengefasst. Man erkennt, dass die Helligkeit vom clippingfreien Modus (oben rechts) über den

⁵ Bei kritischer Betrachtung der Tatsache, dass die LSFs mit dem arithmetischen Mittelwert kondensiert werden (siehe Abschnitt 5.3.1), muss man feststellen, dass im clippingfreien Modus eine geringe Wahrscheinlichkeit von Clipping besteht. Allerdings geschieht dies in einem vernachlässigbar kleinen Maß. Ein Setzen des kondensierten Pixels der LSF auf den minimalen Wert würde clippingfreie Ergebnisse garantieren.

(Max + AMW)/2 Modus (unten links) hin zum Modus des arithmetischen Mittelwertes (unten rechts) immer mehr abnimmt.

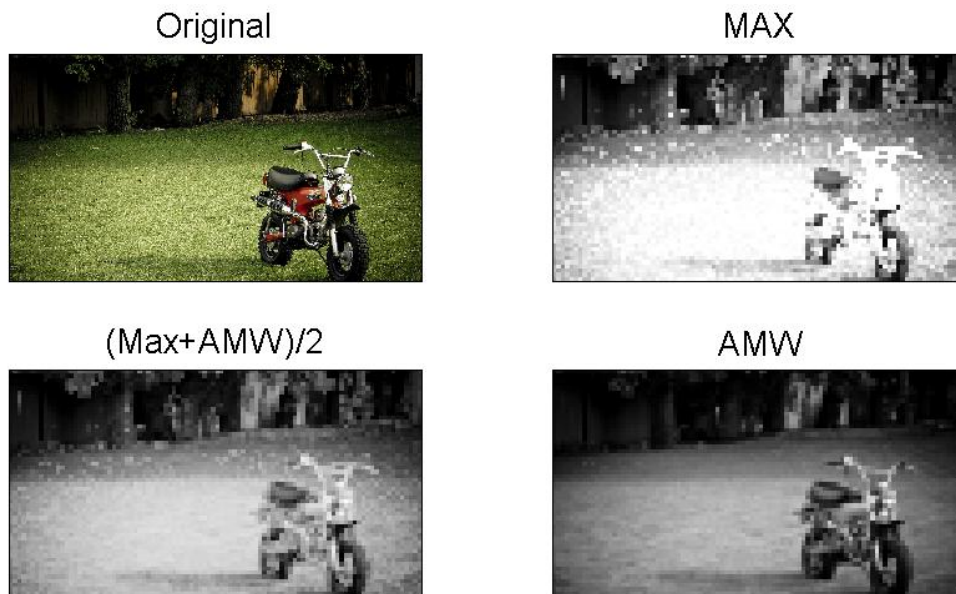


Abbildung 5-7: Vergleich der Kondensierungsfunktionen der starren Condenser-Modi

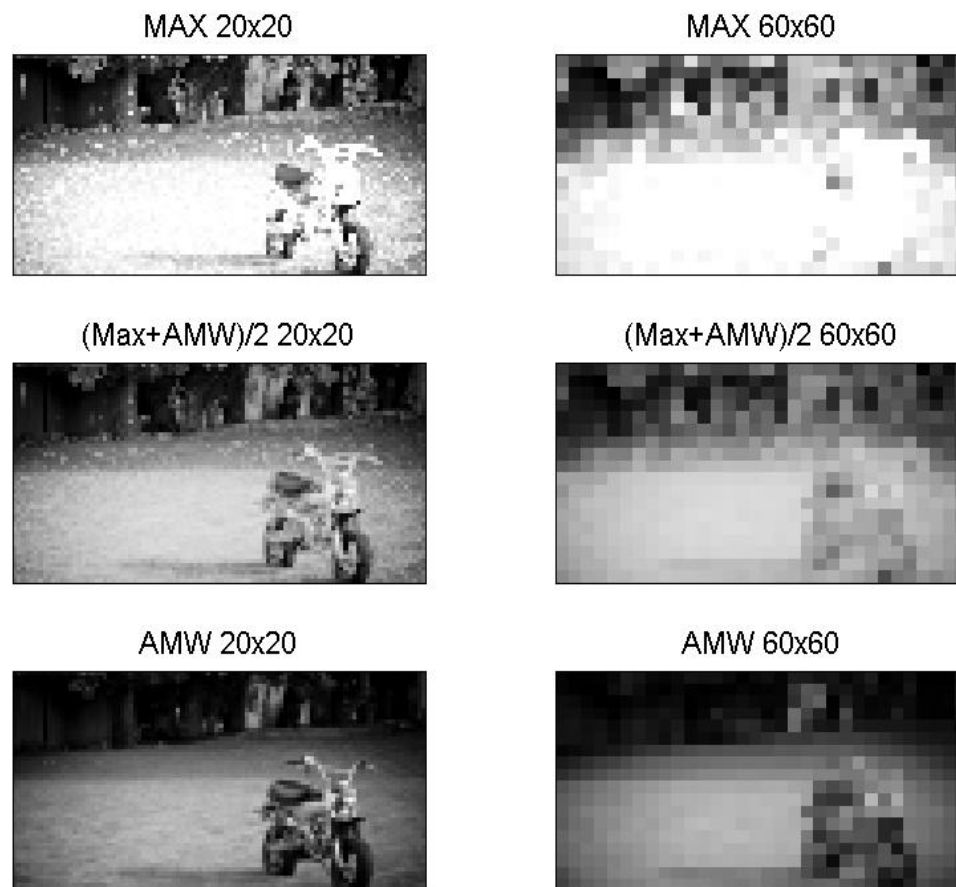


Abbildung 5-8: Vergleich der Zellgröße bei Variation der Kondensierungsfunktion - links: (20·20) Pixel; rechts: (60·60) Pixel

Selbstverständlich hat die Zellgröße ebenfalls einen Einfluss auf das kondensierte Bild. Abbildung 5-8 stellt die oben gezeigten Ergebnisse mit einer Zellgröße von (20·20) Pixeln (linke Seite) denen mit einer Zellgröße von (60·60) Pixeln (rechte Seite) gegenüber.

Beim MAX-Modus (siehe Abbildung 5-8 oben) muss das kondensierte Bild mit steigender Zellgröße heller werden, da hier nur der hellste Pixel verantwortlich für den Wert des kondensierten Pixels ist. Dadurch sinkt die zu erwartende Leistungersparnis. Aufgrund des arithmetischen Mittelwertes sind beim AMW-Modus (siehe Abbildung 5-8 unten) bei steigender Zellgröße manche Bereiche des Bildes dunkler.

Je dunkler das zu prozessierende, kondensierte Bild ist, desto höher ist die zu erwartende Leistungersparnis. Dies bedeutet, dass bei Verwendung der drei oben gezeigten Condenser-Modi durch eine größere maximale Zellgröße mehr Leistung eingespart werden kann.

In Abschnitt 5.4.3 wird jedoch eine Möglichkeit aufgezeigt, wie man auch bei geringer maximaler Zellgröße mehr Leistung einsparen kann.

5.4.2 Automatischer Condenser-Modus

Die Aufgabe des automatischen Modus ist es, die charakteristischen Merkmale der Zellen eines Bildes zu erkennen und den kondensierten Pixel anhand dieser Merkmale durch die Wahl einer geeigneten Kondensierungsfunktion zu bestimmen. Dabei wird die Tatsache ausgenutzt, dass Clipping in verschiedenen Bildbereichen unterschiedlich starke, visuelle Auswirkungen verursacht.

Ziel des automatischen Modus ist es, die Leistungersparnis zu maximieren, ohne dass die visuelle Qualität des prozessierten Bildes verschlechtert wird.

5.4.2.1 Einführendes Beispiel

Im oberen Teil von Abbildung 5-9 ist eine Landschaftsaufnahme zu sehen. Aus dieser werden zwei zu kondensierende Zellen und deren Helligkeitsprofil näher betrachtet.

Zelle 1 zeigt eine Grasfläche, welche am oberen Rand von einer Reihe heller Steine abgeschlossen wird.

Betrachtet man die beiden markierten Helligkeitsprofile der Zelle, so erkennt man, dass die Steinreihe heller ist als die restlichen Pixel der Zelle und dass deren Helligkeitsprofil einem Schwingen zwischen hellen und dunklen Grautönen bei hoher Frequenz ähnelt. Hier führt ein Zulassen von Clipping dazu, dass die Spitzen des Profils (siehe Abbildung 5-9 unten links) abgeschnitten werden. Da jedoch immer nur sehr wenige nebeneinanderliegende Pixel von Clipping betroffen sind und die Dynamik des Bildes erhalten bleibt, hat das Clipping keine negative Auswirkung auf die visuelle Qualität (siehe Abbildung 5-10). In Zelle 1 darf also Leistung eingespart und Clipping zugelassen werden.

Anders verhält es sich in Zelle 2, welche eine Wolke zeigt.

Beim Betrachten des markierten Helligkeitsprofils der Zelle (siehe Abbildung 5-9 unten links) erkennt man, dass der Unterschied zwischen dem hellsten und dem dunkelsten Pixel sehr gering ist. Da der Rest der Zelle dunkler ist als die Wolke, wirkt sich das Clipping unmittelbar auf die Konturen der Wolke aus. Die feinen Abstufungen innerhalb der Wolke gehen verloren, wodurch diese flach und unnatürlich aussieht (siehe Abbildung 5-11). Hier darf keine zusätzliche Leistung eingespart und Clipping verursacht werden, da dieses vom Betrachter als störend wahrgenommen wird.

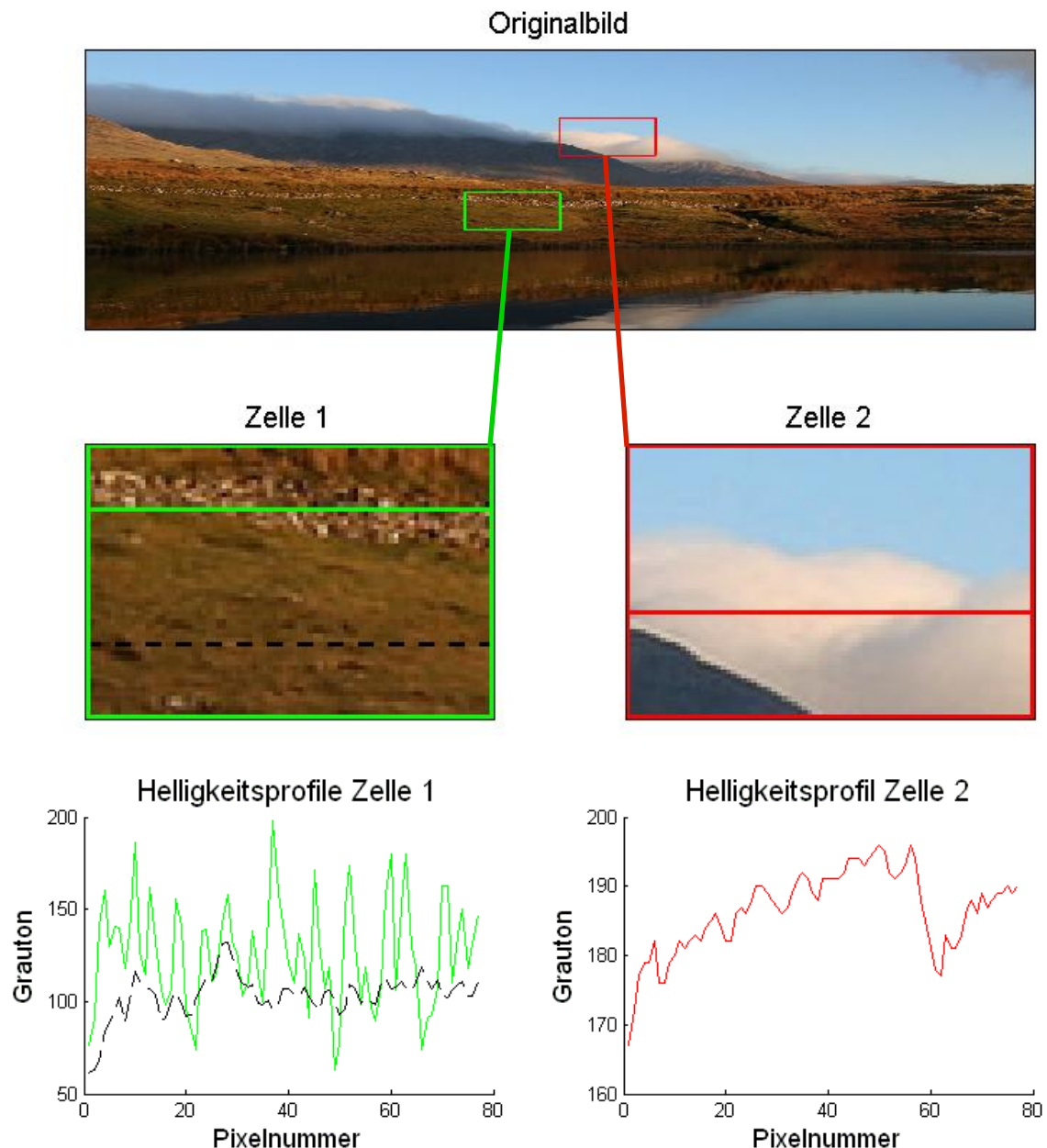


Abbildung 5-9: Beispiel für unterschiedliche Auswirkungen von Clipping in verschiedenen Bildbereichen – oben: Originalbild; Mitte: betrachtete Zellen in vergrößerter Darstellung; unten: Helligkeitsprofil(e) der jeweiligen Zellen

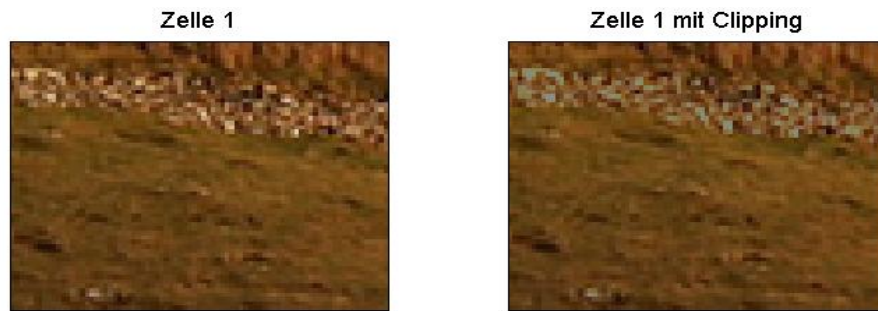


Abbildung 5-10: links: Zelle 1; rechts: Zelle 1 mit Clipping

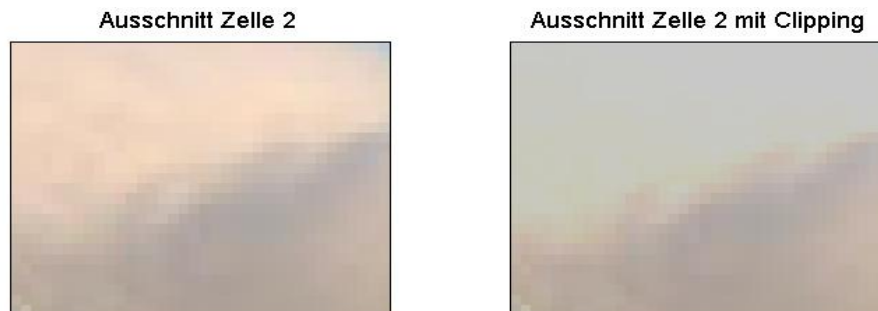


Abbildung 5-11: links: Ausschnitt aus Zelle 2; rechts: Ausschnitt aus Zelle 2 mit Clipping

5.4.2.2 Analyse des Bildes mit dem Clipping Allowance Factor

Das einführende Beispiel aus Abbildung 5-9 hat gezeigt, dass Clipping in verschiedenen Bereichen eines Bildes unterschiedlich starke Auswirkungen auf die Bildqualität haben kann. Im Folgenden wird gezeigt, wie der automatische Modus diesen Sachverhalt durch die Analyse des Inhaltes der zu kondensierenden Zelle erkennt und bei der Kondensierung berücksichtigt.

Zur Bestimmung des Wertes des kondensierten Pixels einer Zelle wird zunächst der sogenannte Clipping Allowance Factor (*CAF*) eingeführt, welcher sich aus einer Funktion von *MAX* und *AMW* der Zelle berechnet.

$$CAF = f(\text{MAX}, \text{AMW}) \quad 5.4$$

Abbildung 5-12 zeigt eine grafische Darstellung des *CAF* aus Formel 5.4.

Für jede Zelle wird der *CAF* berechnet und mit vordefinierten Schwellwerten verglichen. Diesen Schwellwerten werden verschiedene Kondensierungsfunktionen zugeordnet, welche dann den Wert des kondensierten Pixels bestimmen.

Abbildung 5-13 zeigt ein Beispiel für eine automatische Kondensierung. Für dieses Beispiel werden vier Schwellwerte vorgegeben. Bei der ersten, der dritten und der fünften der insgesamt fünf Kondensierungsfunktionen, die durch die Schwellwerte voneinander getrennt werden, handelt es sich um die in Abschnitt 5.4.1 gezeigten starren Kondensierungsfunktionen (*MAX*, *AMW* und $(\text{MAX}+\text{AMW})/2$).

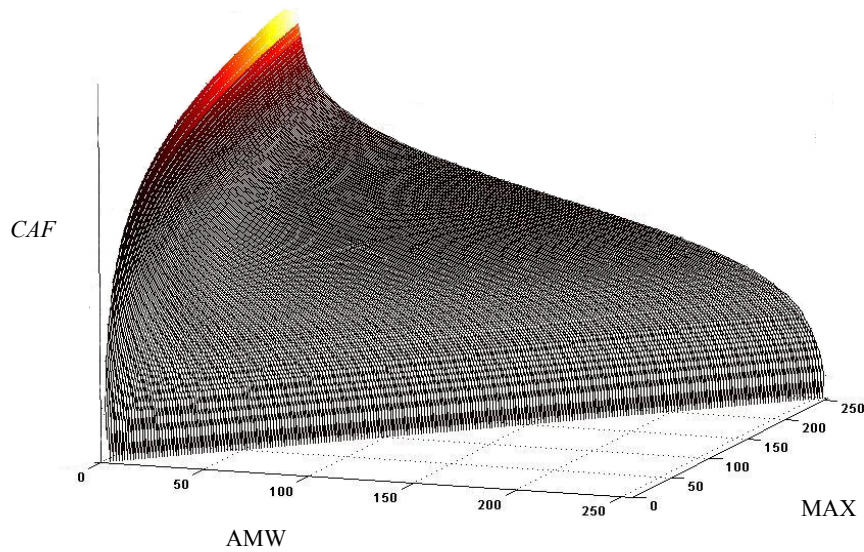


Abbildung 5-12: Grafische Darstellung des CAF

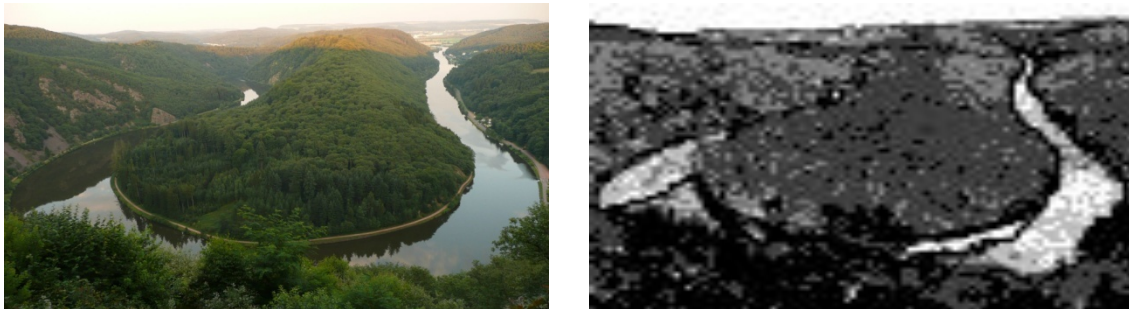


Abbildung 5-13: links: Originalbild; rechts: je nach Kondensierungsfunktion unterschiedlich eingefärbte Zellen

Die gewählte Zellgröße beträgt in diesem Beispiel (20·15) Pixel, die Zellen sind entsprechend ihrer Kondensierungsfunktion unterschiedlich eingefärbt. Wird eine Zelle weiß markiert, bedeutet das, dass hier kein Clipping auftreten darf. Dementsprechend wird der in Abschnitt 5.4.1.1 beschriebene, clippingfreie Modus (MAX) verwendet. Die mittelgrau eingefärbten Zellen erlauben mittleres Clipping. Hier findet die in Abschnitt 5.4.1.2 beschriebene Funktion $(MAX+AMW)/2$ Verwendung. Die schwarz markierten Zellen erlauben viel Clipping, was zu einer hohen Leistungersparnis führt. Die verwendete Funktion ist die des AMW. Bei den hellgrau und den dunkelgrau markierten Zellen liegt die Kondensierungsfunktion zwischen den oben beschriebenen Funktionen.

Man erkennt, dass die meisten Zellen eine mittlere Leistungersparnis, dass viele Zellen eine hohe Ersparnis und dass nur wenige weiße Zellen eine geringe Ersparnis zulassen.

Wie in diesem Beispiel zu sehen ist, werden benachbarten Zellen oftmals unterschiedliche Kondensierungsfunktionen zugeordnet. Dies führt jedoch zu keinen visuellen Artefakten, da Zellen mit ähnlichem Zellinhalt identische Kondensierungsfunktionen zugewiesen bekommen.

Es bietet sich auch die Verwendung zweier verschiedener automatischer Modi an, wobei der erste Modus für unbewegte und der zweite Modus für bewegte Bilder vorgesehen ist. Der einzige Unterschied der beiden Modi besteht dann in verschiedenen vordefinierten

Schwellwerten. Bei bewegten Bildern ist das Auge bezüglich Details weniger empfindlich, daher können die Schwellwerte hier so gewählt werden, dass noch mehr Leistung eingespart wird. Bei unbewegten Bildern muss ein wenig mehr Helligkeit und somit Leistung spendiert werden, um dem Betrachter eine gute Bildqualität zu bieten.

Die Schwellwerte hängen jedoch nicht nur davon ab, ob es sich um ein bewegtes oder ein unbewegtes Bild handelt. Sie werden ebenfalls durch die Eigenschaften und den Typ des Backlights bestimmt. So beeinflusst jede LED eines Edge-Lit Backlights mehr Pixel als eine LED eines Direct-Lit Backlights. Daraus folgt, dass bei einem Edge-Lit Backlight der Wert des kondensierten Pixels etwas niedriger angesetzt werden kann als bei einem Direct-Lit Backlight, da mehr Überschuss an Helligkeit durch die übrigen LEDs erzeugt wird.

Darüber hinaus kann die Anzahl der verwendeten Schwellwerte und der daraus resultierenden Kondensierungsfunktionen variiert werden. Eine höhere Anzahl von Schwellwerten bedeutet, dass das kondensierte Bild präziser an das Originalbild angepasst werden kann. Dadurch kann das Verhältnis aus Leistungersparnis und Bildqualität weiter optimiert werden. Für die meisten Applikationen hat sich die Wahl von vier verschiedenen Schwellwerten bewährt.

5.4.3 Surcondenser

In Abschnitt 5.3 wurde festgestellt, dass die maximale Zellgröße durch die Beschaffenheit des Backlights bestimmt wird. Darüber hinaus wurde im Beispiel aus Abbildung 5-8 gezeigt, dass eine geringe maximale Zellgröße auch eine geringere Leistungersparnis zur Folge haben kann.

Der Surcondenser stellt eine Möglichkeit dar, wie auch mit einer geringen maximalen Zellgröße größere Bildausschnitte betrachtet werden können, um eine noch höhere Leistungersparnis zu erzielen. Beim Surcondenser handelt es sich um eine Erweiterung des Condensers. Wie der Name vermuten lässt, umschließt die vom Surcondenser kondensierte Zelle die des Condensers. Abbildung 5-14 veranschaulicht dies grafisch.

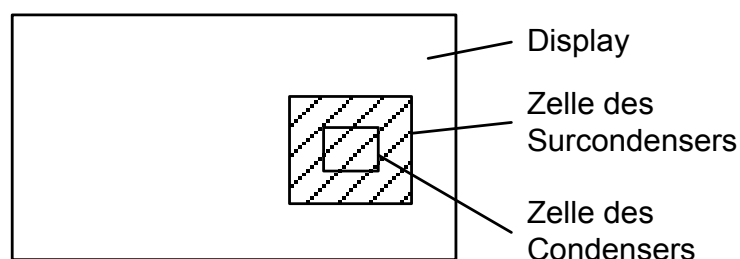


Abbildung 5-14: Zelle des Surcondensers umschließt Zelle des Condensers

Die Zelle der Größe $(s't')$, die vom Surcondenser zu einem Pixel i''_p kondensiert wird, unterliegt, im Gegensatz zur Zelle des Condensers, keiner Einschränkung bezüglich ihrer Größe. Die Gesamtzahl der kondensierten Pixel des Surcondensers ist identisch mit der des Condensers. Der Wert des kondensierten Pixels, wie er später an SSC übergeben wird, ergibt

sich aus den beiden Werten $i'_{p'}$ und $i''_{p'}$. Folgender Pseudocode beschreibt die Vorgehensweise bei der Berechnung:

```

1  if  $i'_{p'} < i''_{p'}$ 
2    do  $i'_{p'} \leftarrow i'_{p'}$ 
3  else if  $i'_{p'} = i''_{p'}$ 
4    do  $i' \leftarrow i'_{p'}$ 
5  else if  $i'_{p'} > i''_{p'}$ 
6    do  $i'_{p'} \leftarrow f(i'_{p'}, i''_{p'})$ 

```

Die beiden Werte $i'_{p'}$ und $i''_{p'}$ werden miteinander verglichen. Falls der Wert des Surcondensers höher oder gleich dem Wert des Condensers ist, wird der Wert des Condensers nicht verändert. Ansonsten wird der Wert des Condensers aus einer Funktion $f(i'_{p'}, i''_{p'})$ ermittelt.

Bei der Wahl der Funktion gibt es wiederum verschiedene Möglichkeiten. Je mehr Leistung eingespart werden soll, desto näher sollte der resultierende Wert an $i''_{p'}$ liegen. Je weniger Clipping produziert werden soll, desto mehr sollte man sich $i'_{p'}$ annähern.

Der kondensierte Pixel des Surcondensers entspricht immer dem AMW der Zelle. Da die Zelle des Surcondensers jedoch über die Pixel des Displays herausragen kann, werden nun zwei Möglichkeiten aufgezeigt, diese Überlappung in die Berechnung mit einzubeziehen.

Cut-Off-Modus:

Die erste Möglichkeit ist, die Zelle des Surcondensers an den Rändern des Displays abzuschneiden. Dadurch verringert sich die Zellgröße in den Randbereichen. Der arithmetische Mittelwert ergibt sich ausschließlich aus den Pixeln des Bildes.

Black-Extended-Modus:

In diesem Modus nutzt man aus, dass die Helligkeit im Randbereich eines Bildes abgesenkt werden kann, ohne dass dies vom Betrachter als störend empfunden wird. Durch die Verwendung des Black-Extended-Modus wird die Überlappung mit schwarzen Pixeln gefüllt, wodurch ein schwarzer Rahmen entsteht. Die Anzahl der Pixel einer Zelle des Surcondensers bleibt in diesem Modus auch an den Rändern des Displays gleich.

Abbildung 5-15 vergleicht die Ergebnisse der Kondensierung nur mit Condenser (oben rechts) mit den Ergebnissen der Kondensierung mit den beiden Surcondenser-Modi (unten). Die gewählte Zellgröße beträgt (60·60) Pixel für den Condenser und (200·200) Pixel für den Surcondenser.

Beim Cut-Off-Modus fällt auf, dass helle Pixel, speziell in der Mitte des Bildes, im Vergleich zum reinen Condenser etwas dunkler dargestellt werden. Beim Black-Extended-Modus werden zusätzlich die Pixel am Rand des kondensierten Bildes abgedunkelt.

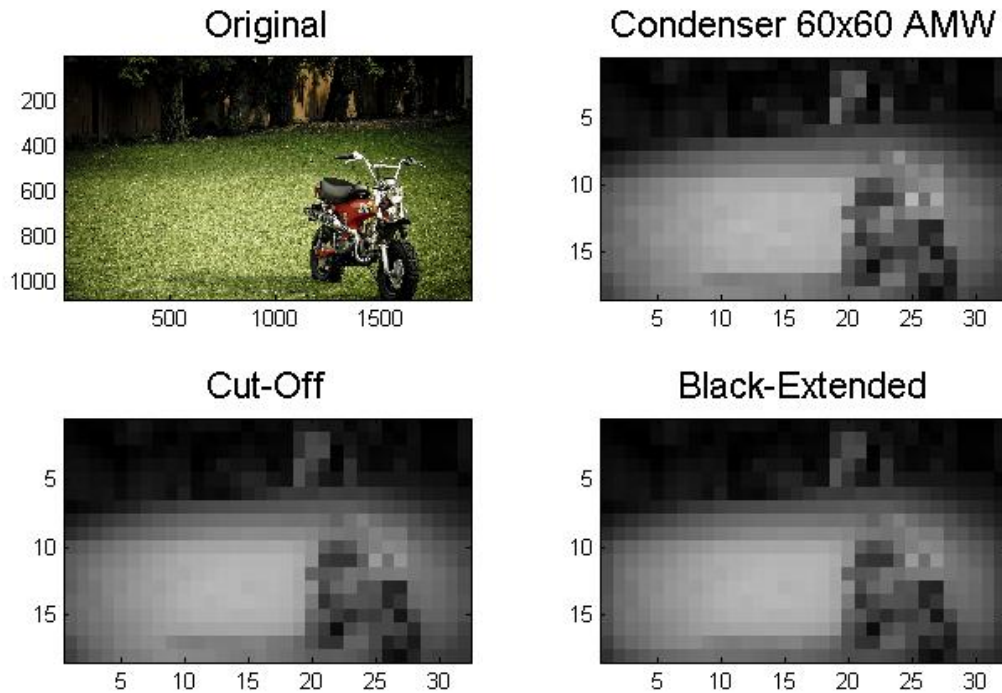


Abbildung 5-15: Vergleich der Ergebnisse des Condensers (oben rechts) mit denen der beiden Surcondenser-Modi (unten)

5.5 Fazit

Da die Laufzeit von SSC proportional zur Anzahl der zu prozessierenden Pixel ist, kann mit dem Condenser der Hardwareaufwand um ein Vielfaches reduziert werden. Bei einem Direct-Lit Backlight führt ein Zusammenfassen von beispielsweise (20·20) Pixeln zu einem kondensierten Pixel zu einer Reduktion der Laufzeit auf 0,25%. Bei einem Edge-Lit Backlight wird beim Zusammenfassen von beispielsweise (60·60) Pixeln zu einem kondensierten Pixel die Laufzeit auf 0,28% reduziert.

Darüber hinaus kann durch die Verwendung des Condensers die Leistungersparnis erhöht werden. Die verschiedenen beschriebenen Kondensierungsfunktionen erlauben sowohl Ergebnisse frei von Clipping als auch Ergebnisse, die clippingbehaftet sind.

Mit dem vorgestellten automatischen Modus kann durch eine Analyse des Zellinhaltes die Leistungersparnis maximiert werden, ohne dass dies Einfluss auf die visuelle Qualität des dargestellten Bildes hat.

Zur Berechnung des kondensierten Pixels werden statistische Größen wie der arithmetische Mittelwert und der Maximalwert der zu kondensierenden Zelle verwendet. Durch das Hinzunehmen des Surcondensers fließt der Mittelwert einer größeren, umschließenden Zelle mit in die Berechnung des kondensierten Pixels ein.

Die Verknüpfung der statistischen Werte der unterschiedlich großen Zellen entspricht in einer einfachen Form einer Frequenzanalyse. Somit wird der Zellinhalt sowohl statistisch als auch bezüglich seiner Frequenz analysiert.

Gerade für den automatischen Modus wäre auch die Berücksichtigung weiterer statistischer Werte unterschiedlich großer Zellen denkbar, um eine differenziertere Analyse des Frequenzspektrums der Zellen durchführen zu können.

Darüber hinaus könnten weitere statistische Größen (Analyse des Histogramms der Zelle - siehe Abschnitt 10.3) zur Bestimmung des kondensierten Pixels berücksichtigt werden.

Der Condenser wirkt sich auf das Originalbild ähnlich wie ein Tiefpassfilter aus. Allerdings bietet er gegenüber einem Standardtiefpassfilter, wie es im Stand der Technik Verwendung findet (siehe Abschnitt 2.3.2.1), drei erhebliche Vorteile:

Der Berechnungsaufwand zur Bestimmung des kondensierten Bildes ist wesentlich geringer. Des Weiteren kann über die Kondensierungsfunktion und die Zellgröße viel differenzierter auf Bilddetails eingegangen werden. Schließlich ist die Auflösung des kondensierten Bildes deutlich höher als die des Backlights.

Das Prozessieren der kondensierten Bilder führt zu Ergebnissen mit perfekter visueller Qualität, zu einer hohen Leistungsparsnis oder zu einem Kompromiss aus beidem. Eine ausführliche Statistik zur Leistungsparsnis bei Variation verschiedener Condenser-Modi erfolgt in Abschnitt 8.2. Die visuelle Qualität der prozessierten Bilder wird in Abschnitt 8.3 diskutiert.

6 Postprozessierung

6.1 Einleitung

Abbildung 6-1 zeigt die gesamte Local-Dimming-Prozessierung in einer Übersicht.

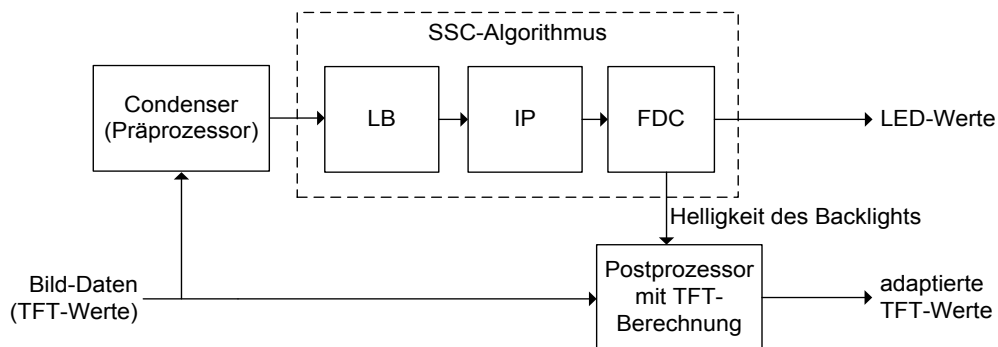


Abbildung 6-1: Übersicht Local-Dimming-Prozessierung

Analog zu Kapitel 5 werden zuerst die Bilddaten kondensiert. Anhand des kondensierten Bildes berechnet der SSC-Prozessor (siehe Kapitel 4) die LED-Werte. Diese werden zum einen zu den LED-Treibern weitergeleitet und zum anderen verwendet, um die erreichte Helligkeit B des Backlights zu berechnen (siehe Formel 3.3). Aufgabe des Postprozessors ist es, den TFT-Wert an das gedimmte Backlight anzupassen. In den meisten Fällen liegt die erreichte Helligkeit in einem Pixel über der vom Originalbild geforderten Helligkeit. Für den Fall eines Helligkeitsdefizites besteht darüber hinaus die Möglichkeit, dieses zu kompensieren und somit die visuelle Qualität des prozessierten Bildes zu erhalten.

Im Folgenden wird erklärt, wie die Adaption der TFT-Werte durchgeführt wird. Dabei wird sowohl die Vorgehensweise bei einer linearen als auch bei einer nichtlinearen TFT-Kennlinie behandelt. In einigen Fällen kann, bedingt durch die Manipulation des Originalbildes im Condenser, ein Helligkeitsdefizit im betrachteten Pixel vorliegen. Dieses Defizit, in der Fachwelt als Clipping bezeichnet, wird in Abschnitt 6.3 näher betrachtet. Dabei wird zwischen Hard- und Soft-Clipping unterschieden. Über eine einfache Berechnung wird gezeigt, wie mit Hilfe des Soft-Clippings entstandene Helligkeitsdefizite kompensiert werden können.

6.2 Berechnung der TFT-Werte

Abbildung 6-2 zeigt den Unterschied zwischen konventionellem und gedimmtem Backlight (oben links) und die daraus resultierende veränderte Transmission der TFT-Zellen (oben rechts) an einem Beispielbild.

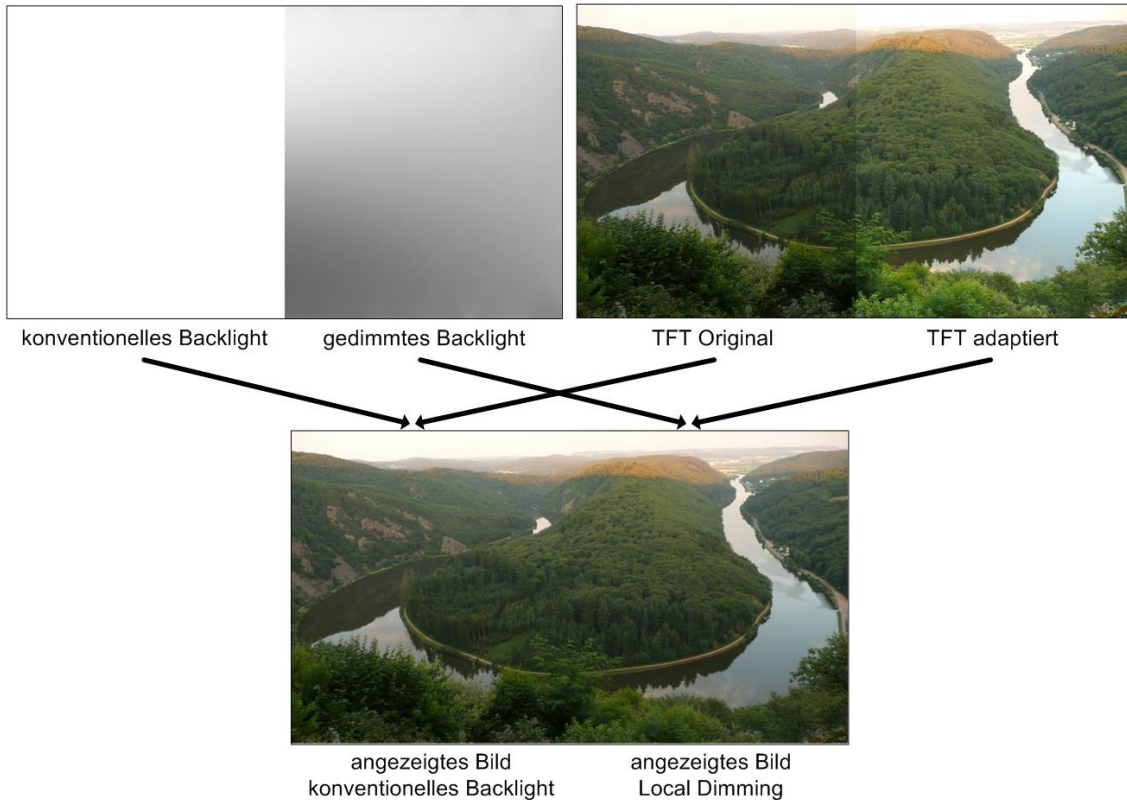


Abbildung 6-2: Vergleich zwischen konventionellem und gedimmtem Backlight - Helligkeit des Backlights (oben links); TFT-Werte (oben rechts); angezeigtes Bild (unten)

Man erkennt, dass die erreichte Helligkeit des lokal gedimmten Backlights individuell an das darzustellende Bild angepasst wird (siehe Abbildung 6-2 oben links). Daher müssen auch die TFT-Werte neu angepasst werden (siehe Abbildung 6-2 oben rechts). Da ein Pixel bei einem Local Dimming Backlight im Allgemeinen weniger stark ausgeleuchtet wird, führt dies dazu, dass die TFT-Werte größer werden müssen, um mehr Helligkeit durchzulassen. Durch die höhere Transmission wird die geringere Helligkeit des Backlights kompensiert. Das angezeigte bzw. vom Betrachter wahrgenommene Bild unterscheidet sich dadurch in der Regel nicht vom Originalbild (siehe Abbildung 6-2 unten).

Im Folgenden werden zwei Fälle für die Berechnung der TFT-Werte unterschieden:

1. Fall: Lineare TFT-Kennlinie

Der adaptierte TFT-Wert t_p für den Pixel p berechnet sich bei linearem Zusammenhang zwischen Helligkeit und eingestelltem TFT-Wert wie folgt:

$$t_p = \min\left(\frac{i_p}{b_p}, 1\right) \quad 6.1$$

Im Fall von Clipping ist das Verhältnis von i_p/b_p größer eins. In diesem Fall wird der TFT-Wert auf sein Maximum gesetzt.

2. Fall: Nichtlineare TFT-Kennlinie

Nach dem Weber-Fechner-Gesetz besteht zwischen der subjektiv empfundenen Stärke und der tatsächlichen Intensität von Sinneseindrücken ein logarithmischer Zusammenhang. Das bedeutet, dass bei exponentiellem Anstieg der physikalischen Helligkeit die vom Auge empfundene Helligkeit nur linear anwächst.

In kommerziellen Fernsehgeräten wird in der Regel die menschliche Wahrnehmung bei der Berechnung der Bilddaten in Form einer Gamma-Korrektur berücksichtigt. Dabei wird die Eingangsgröße in , gemäß Formel 6.2, in eine Ausgangsgröße out überführt.

$$out = in^\Gamma \quad 6.2$$

Ein typischer Wert für Γ bei Fernsehgeräten ist 2,2. Für eine ausführliche Beschreibung der Gamma-Korrektur siehe [Poy 02].

Der nichtlineare Zusammenhang zwischen eingestelltem TFT-Wert und durchgelassener Helligkeit im Pixel wird in einer LUT, der sogenannten Gamma-Tabelle, gespeichert. Aufgrund der Nichtlinearität kann Formel 6.1 nicht mehr zur Berechnung angewendet werden. Die Berechnung der TFT-Werte erfolgt in mehreren Schritten.

Zuerst wird das Verhältnis γ aus der Soll-Helligkeit i_p im Pixel und der aktuell erreichten Helligkeit b_p in diesem Pixel gebildet.

$$\gamma = \min\left(\frac{i_p}{b_p}, 1\right) \quad 6.3$$

Für den Fall, dass Clipping auftritt, ist das Verhältnis von i_p/b_p größer als eins. In diesem speziellen Fall wird γ auf eins gesetzt. Daraufhin wird in der Gamma-Tabelle der dem Verhältnis γ entsprechende TFT-Wert herausgesucht.

$$t_p = \text{LUT} [\gamma] \quad 6.4$$

Die hier beschriebene Vorgehensweise entspricht in etwa der in Abschnitt 2.3.2.2 aufgeführten lokalen Pixelkompensation des algorithmischen Lösungsansatzes von Samsung.

6.3 Clipping

Der Begriff Clipping wird in verschiedenen Disziplinen, wie beispielsweise der Signal- oder der Bildverarbeitung, verwendet. Gemeinhin wird beim Clipping ein Teil des Signals ab einem gewissen Schwellwert abgeschnitten. Im Kontext der vorliegenden Arbeit bedeutet Clipping, dass die geforderte Helligkeit in einem Pixel nicht erreicht wird und damit ein Defizit im Pixel besteht.

Um Clipping zahlenmäßig erfassen zu können, benötigt man eine Berechnungsvorschrift, mit der das entstandene Helligkeitsdefizit in Zahlen ausgedrückt werden kann. Im Kontext dieser Arbeit wird das Helligkeitsdefizit in Grautönen dargestellt. Die Auflösung g der Grautöne entspricht der jeweiligen Auflösung der TFT-Werte. Die verwendete Berechnungsvorschrift für das Clipping c_p im Pixel p lautet:

$$c_p = \left(1 - \frac{i_p}{b_p}\right) \cdot 2^g - 1 \quad 6.5$$

Abbildung 6-3 zeigt den Querschnitt eines Helligkeitsprofils mit auftretendem Clipping. Zur Vereinfachung der Darstellung werden bei dem nun folgenden Beispiel die Helligkeiten als Grautöne angegeben.

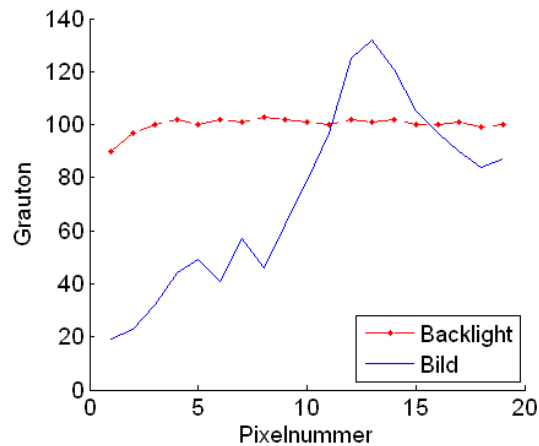


Abbildung 6-3: Beispiel Clipping

Der Pixelwert des Backlights ist in dem betrachteten Beispiel nahezu konstant. Man erkennt, dass die meisten Pixel ausreichend ausgeleuchtet sind. Jedoch liegt zwischen Pixel 12 und 16 ein Helligkeitsdefizit vor.

Im Folgenden werden zwei Arten von Clipping, das harte Clipping (Hard Clipping) und das weiche Clipping (Soft Clipping), unterschieden. Basierend auf Abbildung 6-3 veranschaulicht Abbildung 6-4 diese beiden Arten des Clippings grafisch.

Der linke Teil der Abbildung zeigt den Fall, in dem Hard Clipping auftritt. Die Werte der Pixel 13 bis 16 werden abgeschnitten (geclippt). Das daraus resultierende Bild verliert an Detailgenauigkeit. Hierbei gilt, je größer der mit Hard Clipping behaftete Bildausschnitt und je größer dabei die Differenz zwischen Soll und Ist, desto störender die Auswirkungen auf den Betrachter.

Durch diese Art der Postprozessierung entsteht kein zusätzlicher Hardwareaufwand, da die Bilddaten nicht manipuliert werden müssen.

Anders verhält es sich beim Soft Clipping (siehe rechte Seite der Abbildung 6-4). Hier wird die visuelle Qualität des prozessierten Bildes verbessert. Die erreichte Helligkeit des Backlights ist identisch mit der Helligkeit des Backlights im Beispiel zu Hard Clipping. Allerdings wird der Bildinhalt an die vorhandene Helligkeit adaptiert und die Pixelwerte des Bildes werden anteilmäßig reduziert, woraus eine Verbesserung der visuellen Qualität des Bildes resultiert.

Je weiter jedoch Soll und Ist auseinander liegen, desto schwieriger ist es, der Dynamik des darzustellenden Bildes gerecht zu werden. Des Weiteren verliert man beim Soft-Clipping auch an den Stellen ohne Defizit (Pixel 1 bis 12, Pixel 17 bis 20) an Helligkeit. Das prozessierte Bild erscheint insgesamt dunkler.

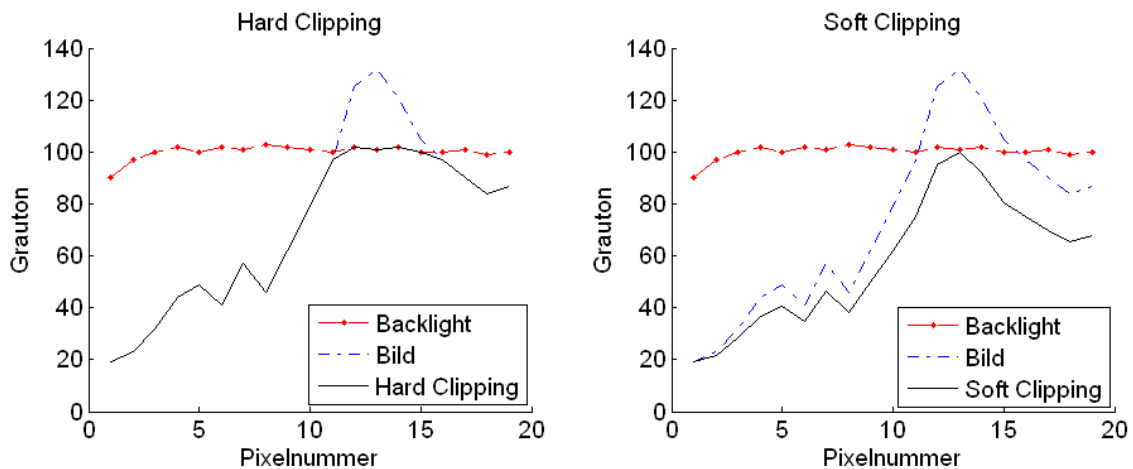


Abbildung 6-4: links: Beispiel Hard Clipping; rechts: Beispiel Soft Clipping

6.4 Methode zum Erhalt der Bildqualität beim Soft Clipping

Ein globaler Soft-Clipping-Ansatz kann auch an Stellen, in denen kein oder nur wenig Clipping auftritt, zu einem signifikanten Verlust an Helligkeit führen. Daher sollte das Herabsetzen der Pixelwerte lokal, also nur in unmittelbarer Umgebung zu den clippingbehafteten Pixeln erfolgen.

Da dem Autor keine Soft-Clipping-Lösung für diesen Fall bekannt ist, wird im Folgenden eine selbst erdachte, einfache Lösung präsentiert, um eine Abschätzung des zusätzlich entstehenden Rechenaufwandes zu erhalten.

Das komplette Bild wird in Gebiete von $(s^{soft} \cdot t^{soft})$ Pixeln partitioniert. Als Randbedingung gilt, dass ein Gebiet kleiner ist als die maximale Zellgröße des Condensers ($s^{soft} \leq s$, $t^{soft} \leq t$). Dann ist zu erwarten, dass die Helligkeit des Backlights in jedem Pixel eines Gebietes in etwa gleich ist. Innerhalb eines Gebietes wird durch einen Vergleich der erreichten mit der geforderten Helligkeit festgestellt, ob Clipping auftritt. Ist das der Fall, so werden der minimale und der maximale Pixelwert des Gebietes und die Position des maximalen Defizites innerhalb des Gebietes ermittelt. Anhand dieser Werte wird der Adaptionfaktor α berechnet (siehe Formel 6.6).

$$\alpha = \frac{\text{Max} - \text{Min}}{b(\text{posMAX}) - \text{Min}} \quad 6.6$$

Die Werte der Pixel des Gebietes werden daraufhin, anhand von Formel 6.7, zu den Soft-Clipping-Werten i_p^{soft} umgewandelt:

$$i_p^{soft} = \alpha \cdot i_p + (1 - \alpha) \cdot \text{Min} \quad 6.7$$

Abschließend werden die TFT-Werte aus b und i_p^{soft} (siehe Abschnitt 6.2) berechnet.

Der folgende Pseudocode fasst noch einmal die oben beschriebene Vorgehensweise zusammen.

```

1  defizit ← 0
2  minimum ← Pixel_max
3  maximum ← 0
4  for lvi ← 1 to s_soft
5    do for lvj ← 1 to t_soft
6      do if defizit < (i(lvi,lvj) - b(lvi,lvj))
7        do defizit ← i(lvi,lvj) - b(lvi,lvj)
8          pos_i ← lvi
9          pos_j ← lvj
10     do if minimum < i(lvi,lvj)
11       do minimum ← i(lvi,lvj)
12     do if maximum < i(lvi,lvj)
13       do maximum ← i(lvi,lvj)
14  diff_alt ← maximum - minimum
15  diff_neu ← b(pos_p) - minimum
16  alpha ← diff_neu - diff_alt
17  if defizit > 0
18    do for lvi ← 1 to s_soft
19      do for lvj ← 1 to t_soft
20        r_neu(lvi,lvj) ← i(lvi,lvj) * alpha + minimum * (1-alpha)

```

Das Ergebnis des eben beschriebenen Verfahrens zur Berechnung von Soft Clipping wird nun mit einer Hard-Clipping-Lösung anhand eines Beispielbildes verglichen (siehe Abbildung 6-5).

Man erkennt, dass die Hard-Clipping-Lösung (siehe linke Seite der Abbildung 6-5) weniger Detailgenauigkeit aufweist als die Soft-Clipping-Lösung (siehe rechte Seite der Abbildung 6-5). So wirken hier beispielsweise die Baumkronen im unteren Teil des Bildes etwas blumenkohlartig. Bei der Soft-Clipping-Lösung bleiben die Details besser erhalten, jedoch fallen die in der Bildmitte auftretenden Kanten beim Übergang von einem Gebiet zum anderen

unangenehm auf. Des Weiteren geht hier im unteren und im rechten Teil des Bildes sehr viel Helligkeit verloren.

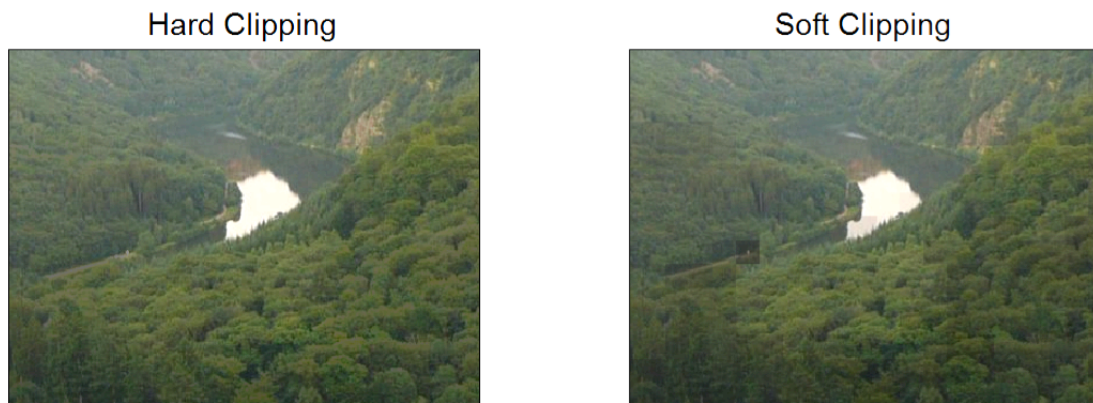


Abbildung 6-5: Vergleich zwischen Hard-Clipping-Lösung (links) und Soft-Clipping-Lösung (rechts)

Die visuelle Qualität des mit Soft Clipping prozessierten Bildes wird zwar lokal aufgebessert, die entstehenden Artefakte werden jedoch als störend wahrgenommen. Der zusätzliche Hardwareaufwand ist, trotz einfachster Vorgehensweise, immens. So müssen in dem hier gezeigten Beispiel beim Auftreten von Clipping innerhalb eines Gebietes 400 Pixel manipuliert werden.

6.5 Fazit

Die Vorgehensweise bei der Postprozessierung ist für jeden Pixel des Displays identisch. Die erreichte Helligkeit wird berechnet und mit der Bildhelligkeit verglichen. Danach werden die TFT-Werte so angepasst, dass der Überschuss an Helligkeit herausgefiltert wird.

Für den Fall, dass Clipping auftritt, bestehen zwei Möglichkeiten, mit diesem umzugehen:

Entweder das Helligkeitsdefizit in einem Pixel wird ohne Weiteres in Kauf genommen (Hard Clipping) oder die Helligkeit des Bildes wird um die clippingbehafteten Pixel herum graduell reduziert (Soft Clipping).

Da beim Soft Clipping möglichst viel Helligkeit erhalten bleiben soll, darf ein Herabsetzen der zu erreichenden Pixelwerte nur in unmittelbarer Umgebung zu den clippingbehafteten Pixeln zugelassen werden.

Die hier gezeigte simple Postprozessierung zum Erhalten von Soft Clipping bedarf jedoch bereits eines immensen zusätzlichen Hardwareaufwandes und führt dennoch zu keinen zufriedenstellenden visuellen Ergebnissen.

Aus diesem Grund ist es fragwürdig, ob der zusätzliche Hardwareaufwand von Soft Clipping bei den oben gezeigten Nachteilen gerechtfertigt ist. Bei den in Abschnitt 7.4 folgenden Prototypen wird die Postprozessierung daher mit Hard Clipping durchgeführt.

7 HW-Implementierung und Prototypen

7.1 Einleitung

Die effiziente Umsetzung des Local-Dimming-Prozessors in Hardware stellt eine der Anforderungen dar, die an diese Arbeit gestellt wurden. Im Kontext der Arbeit zeichnet sich eine effiziente Hardware durch die folgenden drei Aspekte aus:

Geringe Logikkomplexität

Je geringer die Komplexität der Logik, desto kleiner ist die bei der Implementierung benötigte Chipfläche und desto geringer sind die entstehenden Kosten. Eine geringe Logikkomplexität erreicht man durch einfache und wenige Rechenoperationen.

Abschnitt 7.2.2 beschäftigt sich mit der allgemeinen Umsetzung von Condenser, SSC und Postprozessor in Hardware.

Geringer Speicherbedarf

Ebenso wie die Logik verursacht auch der benötigte Speicher zusätzliche Kosten und sollte somit möglichst klein gehalten werden.

Die verschiedenen verwendeten Speicherbausteine sowie die Reduktion der Speichermenge werden in Abschnitt 7.2.1 thematisiert.

Geringer Speicherzugriff

Jeder Zugriff auf den Speicher kann eine Berechnung verzögern. Wenn mehrere Akteure auf einen Speicher zugreifen, dann müssen diese so koordiniert werden, dass keine Zugriffskonflikte entstehen. Daraus können weitere Verzögerungen resultieren.

Die Integration der gesamten Local Dimming Hardware in eine bestehende Displaysteuerung ist Gegenstand von Abschnitt 7.3.

Im Folgenden wird zunächst der allgemeine Aufbau der Local Dimming Hardware beschrieben. Diese umfasst, neben den bereits weiter oben erläuterten Komponenten Condenser, SSC und Postprozessor, auch verschiedene Speicherbausteine. Danach wird gezeigt, wie die hier vorgestellte Hardware in eine bestehende Displaysteuerung (vergleiche Abschnitt 2.2.2) integriert werden kann. Basierend darauf werden drei in Hardware implementierte Prototypen für die Berechnung von Local Dimming Backlight präsentiert. Abgeschlossen wird dieses Kapitel mit einem Fazit.

7.2 Allgemeiner Aufbau der Local Dimming Hardware

Das Blockschaltbild der gesamten Local-Dimming-Prozessierung wurde bereits in Abbildung 6-1 gezeigt. Für eine Implementierung in Hardware benötigt man neben den dort gezeigten Komponenten zusätzliche Speicherbausteine. Abbildung 7-1 zeigt den allgemeinen Aufbau der Local Dimming Hardware.

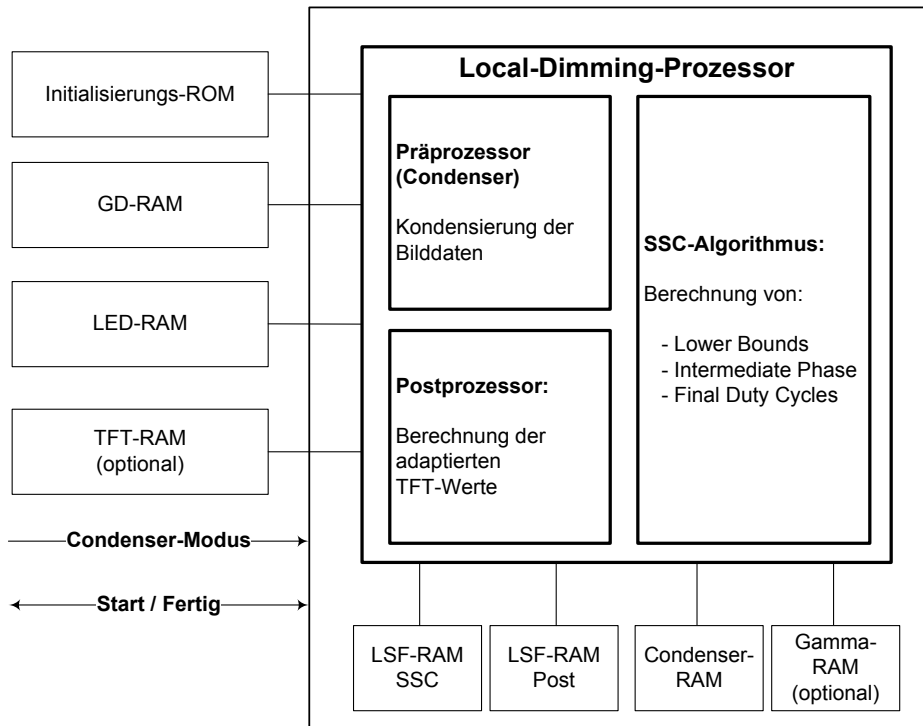


Abbildung 7-1: Allgemeiner Aufbau der Local Dimming Hardware

Der HW-Block benötigt neben dem GD-RAM und dem optionalen TFT-RAM nur einige wenige Schnittstellen zur Umgebung. Hierzu zählt unter anderem der LED-RAM, welcher die von der Local Dimming Hardware berechneten LED-Werte enthält. Daneben kann von außen vorgeschrieben werden, in welchem Modus der Condenser verwendet wird. Zugriffskonflikte mit der Umgebung können vermieden werden, indem der Hardware über ein externes Startsignal mitgeteilt wird, wann die Prozessierung beginnen soll. Im Gegenzug kann die Local Dimming Hardware über ein Signal mitteilen, wann die Prozessierung beendet ist.

Intern hat der Local-Dimming-Prozessor alleinigen Zugriff auf weitere Speicher. Die Speicherbausteine sind Gegenstand von Abschnitt 7.2.1.

7.2.1 Speicherbausteine

7.2.1.1 GD-RAM

Im GD-RAM werden die Bilddaten gespeichert. Der verwendete Speichertyp ist üblicherweise DDR2-SDRAM und die Größe des RAMs entspricht der Bildgröße, im Beispiel von Full-HD

(1920·1080) Pixel pro Farbe. Jeder Subpixel hat in der Regel eine Auflösung von 8 bis 10bit. Die Busbreite ist abhängig vom verwendeten Speicher, entspricht aber mindestens der Breite, die benötigt wird, um einen kompletten Pixel einzulesen.

7.2.1.2 Condenser-RAM

Wie in Kapitel 5 beschrieben, werden die Bilddaten kondensiert und in einen Speicher, den Condenser-RAM, geschrieben. Die kondensierten Bilddaten werden vom SSC-Algorithmus prozessiert. Gewöhnlich genügt ein preiswerter DDR2-SDRAM. Im Fall einer Implementierung in ein FPGA kann wegen der geringen Größe der kondensierten Bilddaten auch ein integrierter SRAM verwendet werden. An die Busbreite wird seitens von SSC keine besondere Anforderung gestellt.

7.2.1.3 TFT-RAM (optional)

Je nach Anwendung können die im Postprozessor berechneten TFT-Werte in einem DDR2-SDRAM abgelegt oder direkt zur Displaysteuerung weitergeleitet werden. Die TFT-Werte haben typischerweise eine Auflösung von 8 bis 10bit pro Subpixel. Im Allgemeinen ist die Größe des TFT-RAMs also identisch mit der des GD-RAMs.

7.2.1.4 LSF-RAM

Für die Local-Dimming-Berechnung sind zwei LSF-RAMs, einer für SSC und einer für die Postprozessierung, vorgesehen. Dies hat zwei Gründe. Zum einen müssen sowohl SSC als auch der Postprozessor nahezu ununterbrochen auf die LSF-Daten zugreifen. Ein gemeinsamer RAM würde zu Zugriffskonflikten führen bzw. die Prozessierung verlangsamen. Zum anderen unterscheiden sich die Daten voneinander, welche von SSC und dem Postprozessor benötigt werden.

Bei der Prozessierung von SSC werden die LED-Werte mit Hilfe des kondensierten Bildes und der kondensierten LSFs berechnet. In Abschnitt 5.3 wird gezeigt, dass für die Kondensierung der LSFs eine Zelle von $(s \cdot t)$ Pixeln zu einem kondensierten Pixel zusammengefasst wird. Die virtuelle Position des kondensierten Pixels liegt in der Mitte der Zelle (siehe Abbildung 7-2).

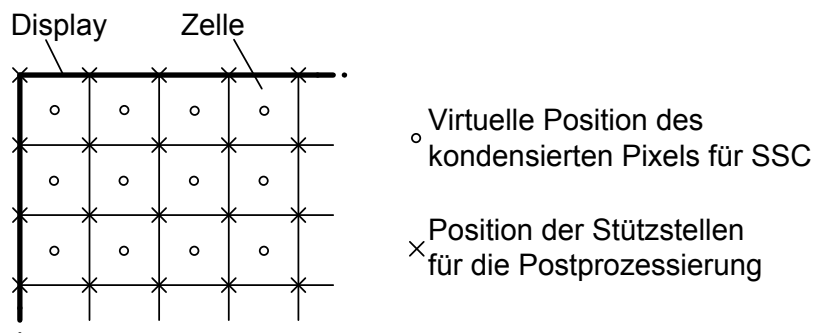


Abbildung 7-2: Virtuelle Position der kondensierten Pixel für SSC und Position der Stützstellen für die Postprozessierung

Verwendet man die kondensierten LSFs für die Postprozessierung, ergeben sich Schwierigkeiten. Es existiert keine Information über die Pixel am Rand des Displays. Deren Werte müssten abgeschätzt werden bzw. zusätzlich abgespeichert werden, damit eine Interpolation möglich ist. Dies würde wiederum zu ungleich großen zu interpolierenden Gebieten führen, wodurch der Ablauf der Postprozessierung erschwert wird.

Daher wird die in Abbildung 7-2 gezeigte Position der Stützstellen für die Postprozessierung verwendet. Der Wert, welcher an den Stützstellen in den LSF-RAM geschrieben wird, entspricht dem der originalen LSF. Zur effizienten Berechnung der Interpolation zwischen zwei Stützstellen (siehe Abschnitt 7.2.2.3) empfiehlt es sich, dass der Abstand der Pixel zwischen diesen eine Zweierpotenz beträgt.

Durch den Condenser und die Verwendung von Stützstellen für die Postprozessierung wird der Speicherbedarf für die LSFs bereits drastisch reduziert. Eine weitere Möglichkeit zur Reduktion des Speicherbedarfs ist die Ausnutzung von Symmetrien innerhalb einer LSF und zwischen unterschiedlichen LSFs. Beim Betrachten des oberen Teiles von Abbildung 4-6 und Abbildung 4-7 fällt auf, dass sich die LSFs verschiedener LEDs oftmals sehr ähnlich sind. Diese Ähnlichkeit kann ausgenutzt werden, um die benötigte Größe des LSF-RAMs weiter zu verringern. Symmetrien können aber nur dann ausgenutzt werden, wenn sich die Helligkeitsverteilungen der einzelnen Dioden bzw. deren LSFs nur gering voneinander unterscheiden (siehe hierzu auch Abschnitt 10.5).

7.2.1.5 Gamma-RAM (optional)

Falls eine nichtlineare Kennlinie zwischen TFT-Wert und Helligkeit vorliegt (siehe Abschnitt 6.2), wird diese im Gamma-RAM hinterlegt. Der Speicherbedarf für den Gamma-RAM ist sehr gering.

7.2.1.6 Initialisierungs-ROM

Da es sich bei den oben vorgestellten Speicherbausteinen um flüchtige Speicher handelt, müssen die darin enthaltenen Daten bei einem Neustart des Systems erst wiederhergestellt werden. Der Initialisierungs-ROM beinhaltet diese Daten. Die Größe des Initialisierungs-ROMs lässt sich durch Summieren aller zur Local Dimming Hardware gehörenden Speicher ermitteln.

7.2.2 Local-Dimming-Prozessor

7.2.2.1 Condenser Hardware

Die Pixel eines Bildes werden, wie in Abbildung 7-3 gezeigt, zeilenweise von links oben nach rechts unten eingelesen und in der Regel auch in dieser Reihenfolge im GD-RAM abgelegt (siehe Abschnitt 7.2.1.1). Die Breite des Datenbusses und die Bittiefe der Pixel können dabei von Display zu Display variieren. Gewöhnlich ist die Busbreite so ausgelegt, dass alle drei Subpixel auf einmal eingelesen werden können.

Die mathematischen Operationen des Condensers sind so einfach und schnell, dass die Kondensierung der Pixel parallel zum Einlesen des Bilddatenstromes durchgeführt werden kann. Bei einem Backlight mit weißen LEDs ist für jeden Pixel nur der hellste Subpixel relevant, welcher durch eine Vergleichsoperation ermittelt wird. Aus allen ($s \cdot t$) hellsten Subpixeln wird der arithmetische Mittelwert der Zelle gebildet. Darüber hinaus muss man sich den höchsten Wert aus den hellsten Subpixeln für jede Zelle merken. Diese beiden Werte pro Zelle reichen aus, um alle der in Abschnitt 5.4 aufgeführten Kondensierungsfunktionen zu verwenden.

Für die Kondensierung von N Pixeln pro Zeile und einer Breite von t Pixeln pro Zelle benötigt man $2 \cdot N/t$ Register, je eines für die Summe der Pixelwerte und eines für den maximalen Pixelwert. Nach jeder s -ten Zeile wird der Wert des kondensierten Pixels aus Mittelwert und Maximalwert errechnet und in den Condenser-RAM (siehe Abschnitt 7.2.1.2) geschrieben.

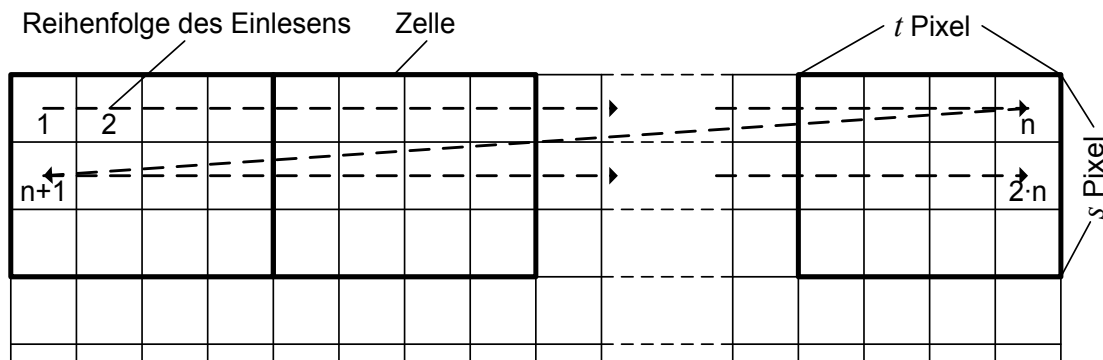


Abbildung 7-3: Reihenfolge beim Einlesen eines Bildes

Falls der in Abschnitt 5.4.3 eingeführte Surcondenser verwendet wird, muss für diesen der arithmetische Mittelwert aus den ($s' \cdot t'$) Pixeln berechnet werden. Für eine Implementierung in Hardware bietet es sich an, dass s' und t' jeweils ungerade ganzzahlige Vielfache von s und t sind. In diesem Fall kann der arithmetische Mittelwert der Surcondenser-Zelle aus den arithmetischen Mittelwerten der Condenser-Zellen berechnet werden. Ansonsten werden zusätzliche Register benötigt.

7.2.2.2 SSC Hardware

SSC benötigt lesenden Zugriff auf den Condenser-RAM und den LSF-RAM und schreibenden Zugriff auf den LED-RAM.

Die verwendeten mathematischen Bauelemente sind Addierer, Subtrahierer und einige wenige Multiplizierer. Da Dividierer sehr aufwändig und auf manchen FPGAs nicht direkt implementiert werden können, ist es möglich, die Divisionen durch Schiebe- und Vergleichsoperationen zu ersetzen (siehe Appendix D – Division durch Schiebe- und Vergleichsoperationen).

Je nach Busbreite und Anzahl der LEDs müssen ein oder mehrere Zugriffe auf den LSF-RAM für die Prozessierung eines Pixels erfolgen. Sobald die ersten Daten vorliegen, werden die

Einflüsse der LEDs mit dem jeweiligen LED-Wert multipliziert. Die den Pixel dominierende LED $x(k)$ wird dabei zunächst nicht berücksichtigt. Dann wird das Defizit Δ_p im Pixel p berechnet:

$$\Delta_p = i_p - \sum_{\ell \neq k} a_p(\ell) \cdot x(\ell) \quad 7.1$$

Im Fall der LB werden die LED-Werte durch die Preset-Werte ersetzt. Im Fall der optionalen IP wird der Pixelwert i_p durch den Offsetwert δ bzw. durch den Sensibilitätsfaktor λ modifiziert.

Über die oben erwähnten Schiebe- und Vergleichsoperationen wird dann der LED-Wert anhand von Formel 7.2 festgelegt:

$$x(k) \geq \frac{\Delta_p}{a_p(k)} \quad 7.2$$

Falls der berechnete Wert für $x(k)$ größer als der bisherige Wert ist, wird dieser neue Wert übernommen.

7.2.2.3 Postprozessor Hardware

Die Postprozessor Hardware benötigt lesenden Zugriff auf den GD-RAM, den LED-RAM und den LSF-RAM. Falls ein TFT-RAM Verwendung findet, so wird dieser von der Postprozessor Hardware beschrieben.

Zunächst werden in einem ersten Schritt die vier Stützstellen aus dem LSF-RAM für die Postprozessierung eingelesen. Mit Hilfe der LED-Werte wird daraus die erreichte Helligkeit des Backlights berechnet. Im Anschluss daran werden in einem zweiten Schritt die übrigen Pixel zuerst vertikal und schließlich Zeile für Zeile in horizontaler Richtung zwischen den Stützstellen linear interpoliert. Der durch die lineare Interpolation entstehende Rundungsfehler ist vernachlässigbar gering. Das Beispiel in Abbildung 7-4 zeigt die drei Schritte der Interpolation für eine Größe von (16·16) Pixeln.

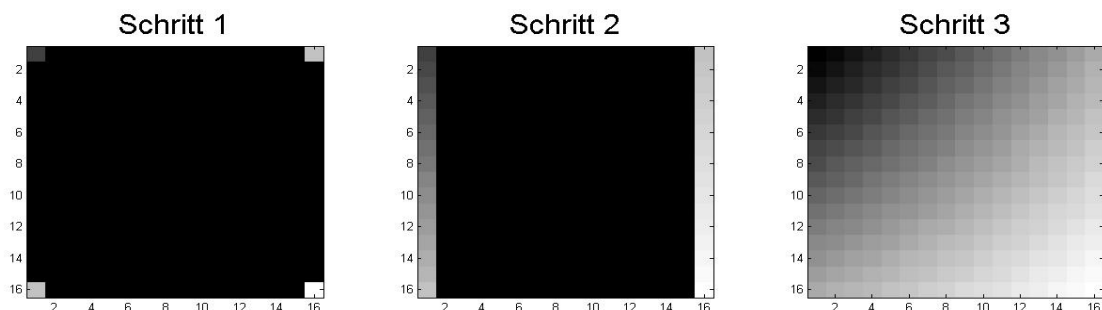


Abbildung 7-4: Vorgehensweise bei der Interpolation der LSFs für die Postprozessierung

Um die Komplexität der Berechnungen möglichst gering zu halten, ist es sinnvoll, dass der Abstand zweier Stützstellen stets eine Zweierpotenz beträgt. Dadurch können die zu

interpolierenden Werte durch Schiebeoperationen statt durch aufwändige Divisionen ermittelt werden.

Zur Berechnung der adaptierten TFT-Werte wird eine der beiden in Abschnitt 6.2 gezeigten Methoden verwendet. Die Division wird, genau wie bei der Berechnung der LED-Werte, ebenfalls durch Vergleichs- und Schiebeoperationen durchgeführt. Die Reihenfolge, mit der die Pixel berechnet werden, entspricht im Allgemeinen der Reihenfolge, in der die Pixel im GD-RAM abgelegt sind – zeilenweise von links oben nach rechts unten.

Weil hier für jeden Pixel exakt die gleichen Berechnungsschritte ausgeführt werden, bietet sich bei der Implementierung eine Pipeline-Struktur an (siehe Appendix F – Pipelining).

7.3 Integration der Local Dimming Hardware in einen bestehenden Displaycontroller

7.3.1 Lage der Local Dimming Hardware

In Kapitel 2.2.2 wurde ein konventioneller Displaycontroller vorgestellt. Es gibt nur eine sinnvolle Möglichkeit, wie man die Local Dimming Hardware in diesen integrieren kann, nämlich zwischen den Videoprozessor und das LCD-Modul (siehe Abbildung 7-5).

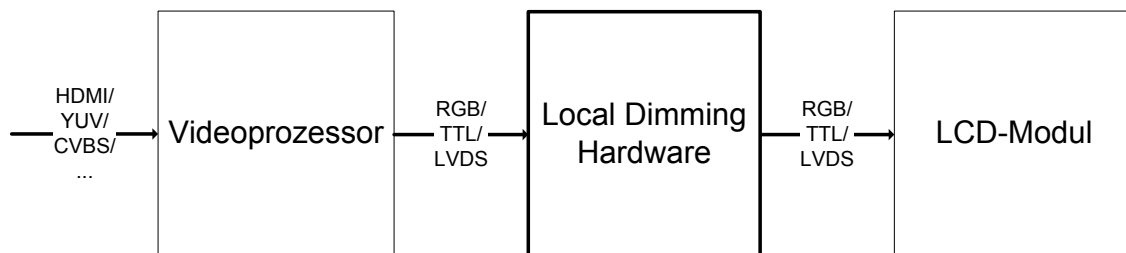


Abbildung 7-5: Architektur des Displaycontrollers mit integrierter Local Dimming Hardware

Dadurch können alle Maßnahmen, die im Videoprozessor zu einer Aufbereitung bzw. Verbesserung der Bilddaten getroffen wurden, genauso beibehalten werden. Die Local Dimming Hardware greift erst dann auf die Bilddaten zu, wenn diese in der vom LCD-Modul benötigten Form vorliegen.

Neben der Synchronisation von Zeilen- und Spaltentreibern kommt dem LCD-Modul eine neue Aufgabe zu: die Synchronisation vom Backlight mit dem darzustellendem Bild.

7.3.2 Pufferung und Manipulation der Bilddaten

Die Bilddaten werden vom Videoprozessor zum LCD-Modul geleitet. Dabei wird in jedem Takt (CLK) ein Pixel gesendet. Die Local Dimming Hardware puffert einige wenige Pixel in einer Warteschlange mit FIFO-Verhalten, welche dann für den Condenser und die Postprozessierung verwendet werden. Dies wird in Abbildung 7-6 verdeutlicht.

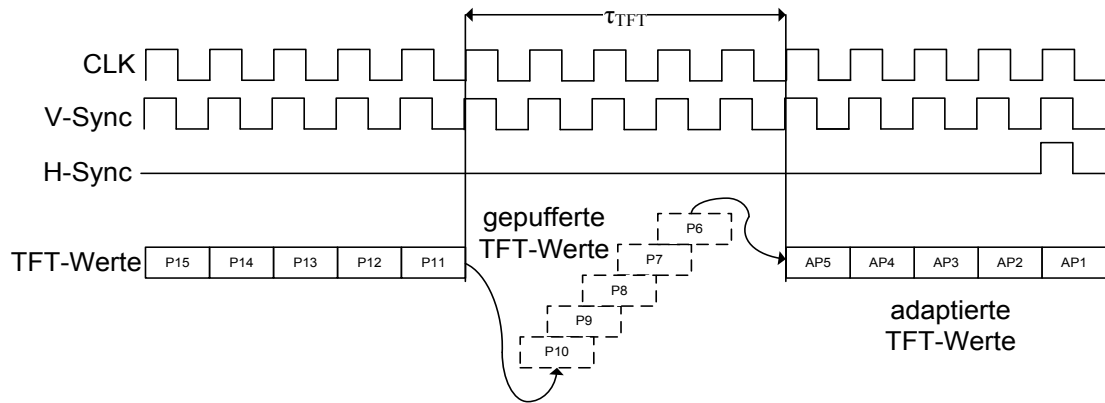


Abbildung 7-6: Pufferung des Bilddatenstromes und der Synchronisationssignale zwischen Videoprocessor und LCD-Modul

Die Warteschlange puffert so viele Pixel, wie nötig sind, um den ersten adaptierten TFT-Wert zu prozessieren. Dabei ist τ_{TFT} die Zeit, die benötigt wird, um einen adaptierten TFT-Wert zu berechnen. τ_{TFT} liegt in einer Größenordnung von 20 bis 30 Takten. Der adaptierte TFT-Wert wird anstelle des ersten Bildpixels an das LCD-Modul weitergegeben. Durch Pipelining wird mit jedem Takt ein weiterer adaptierter TFT-Wert an das LCD-Modul weitergeleitet.

Man erkennt, dass durch die präsentierte Vorgehensweise keine Zugriffskonflikte oder sonstige Einschränkungen bei der Integration der Local Dimming Hardware verursacht werden.

7.3.3 Zeitlicher Ablauf der Prozessierung

Zur Darstellung des zeitlichen Ablaufs der Prozessierung für Videoanwendungen muss der Aufbau des Local-Dimming-Prozessors noch weiter präzisiert werden (siehe Abbildung 7-7).

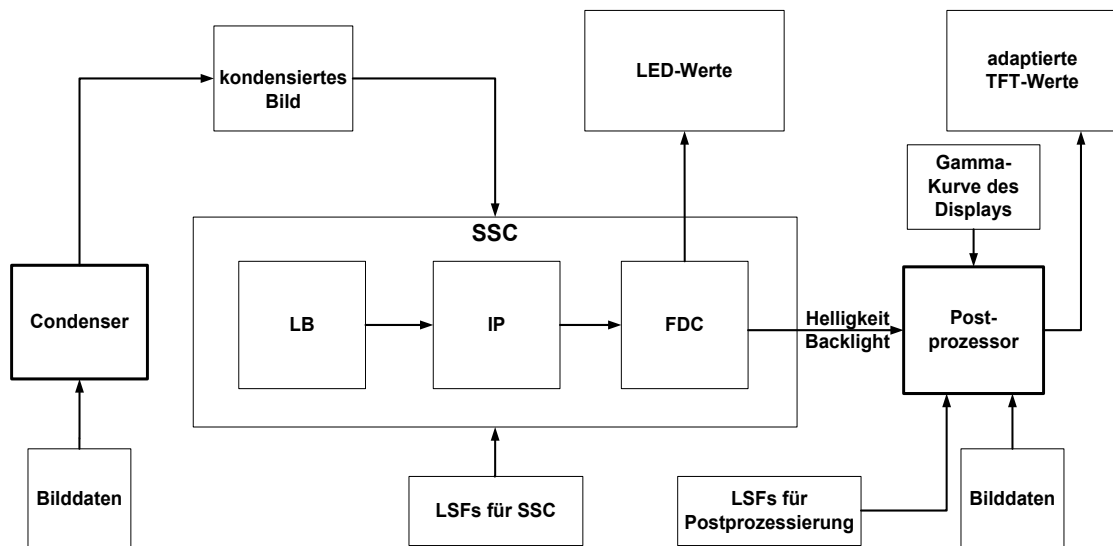


Abbildung 7-7: Präzisiertes Aufbau des Local-Dimming-Prozessors

In Abschnitt 6.3 wird gezeigt, dass mit jedem Takt ein Pixel an das LCD-Modul weitergeleitet wird. Das bedeutet, dass die Zeit, die zum Aufbau eines Frames benötigt wird, proportional zur Auflösung des Displays ist. Gepaart mit einer hohen Bildwiederholrate führt dies dazu, dass der Datenfluss nur für kurze Zeit (zwischen zwei Frames) unterbrochen wird. Die Auswirkungen, die dies auf die Local-Dimming-Prozessierung hat, sind in Abbildung 7-8 zu sehen.

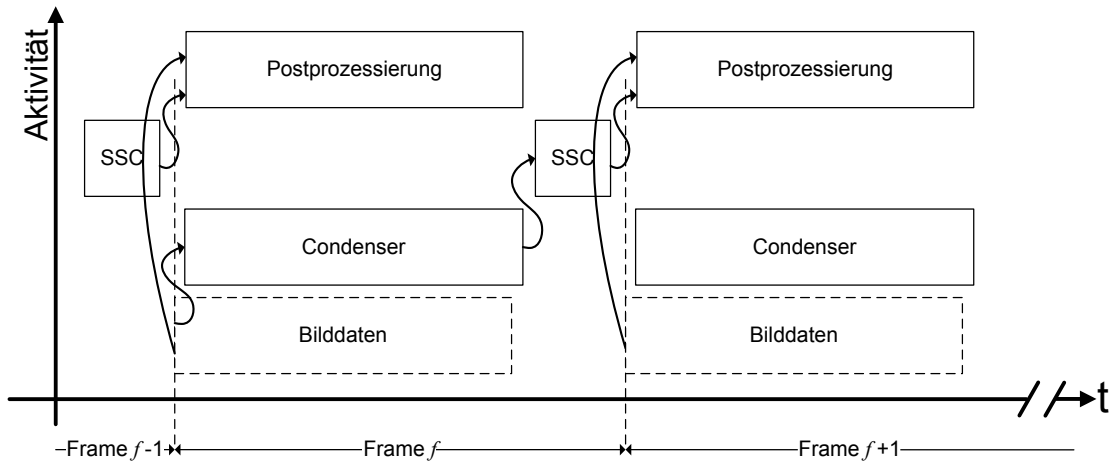


Abbildung 7-8: Timing-Diagramm für Videoanwendung

Zuerst werden die Bilddaten von Frame (f) kondensiert. Die LED-Werte für das kondensierte Frame (f) werden mit SSC berechnet und liegen bereit, bevor das neue Frame ($f+1$) eingelesen wird. Sobald der Datenstrom des neuen Frames ($f+1$) beginnt, wird dieser wiederum kondensiert. Parallel dazu werden die TFT-Werte mit Hilfe der Bilddaten von Frame ($f+1$) und der LED-Werte des vorherigen Frames (f) berechnet. Dieses Vorgehen ist deshalb möglich, da der Unterschied zwischen zwei aufeinanderfolgenden Frames in der Regel sehr gering ist. Das bedeutet, dass sich auch die LED-Werte nur geringfügig voneinander unterscheiden.

Natürlich kann es bei Bildschnitten, schnellen Kamerabewegungen etc. zu größeren Unterschieden zwischen dem Bildinhalt zweier Frames kommen. In diesem Fall besteht die Gefahr, dass die generierte Helligkeit des Backlights zu gering ist. Die hierbei entstehenden möglichen Fehler werden in Abschnitt 8.3.2 untersucht und bewertet.

Möchte man dieses Problem umgehen, so muss ein zusätzlicher Speicher in Größe des GD-RAM bereitgestellt werden. Darin werden die Bilddaten des vorherigen Frames gepuffert und mit den dazu passenden LED-Werten an das LCD-Modul weitergegeben.

7.4 Prototypen

In diesem Abschnitt werden drei verschiedene Prototypen für SSC vorgestellt. Diese dienen als Beweis dafür, dass die Local-Dimming-Prozessierung unter realen Bedingungen eingesetzt werden kann. Darüber hinaus wird anhand der Prototypen gezeigt, dass sich die entwickelte Methode zur Berechnung von Local Dimming Backlight effizient in Hardware implementieren lässt.

Bei der Auswahl der Prototypen wurde darauf geachtet, dass sich diese wesentlich voneinander unterscheiden. Es werden, neben einem Full-HD Edge-Lit TV, zwei Direct-Lit Displays vorgestellt: Ein Full-HD TV und eine Signalanzeige mit geringerer Auflösung. Tabelle 7-1 zeigt dies in Form eines Überblicks der Kenndaten.

| Parameter | Direct-lit TV | Edge-lit TV | Direct-Lit Signalanzeige |
|--------------|---------------|-------------|--------------------------|
| LEDs·Dioden | 180·12 | 6·38 | 72·4 |
| Displaygröße | 42'' | 32'' | [168·30]cm |
| Auflösung | Full-HD | Full-HD | 104·576 Pixel |

Tabelle 7-1: Kenndaten der Prototypen

Der Aufbau der nun folgenden drei Unterkapitel ist nahezu identisch. Zuerst werden die Prototypen kurz vorgestellt und deren LSFs und Einflussmatrix präsentiert. Wie man anhand von Mess- bzw. Simulationsdaten auf eine LSF bzw. die Einflussmatrix kommt, ist Gegenstand von Appendix G. Anhand der Einflussmatrix werden Rückschlüsse auf die Local Dimming Hardware geschlossen, welche dann im Folgenden in aller Kürze beschrieben wird. Anschließend werden die Ergebnisse der Implementierung vorgestellt. Zusätzlich wird bei den letzten beiden Prototypen gezeigt, wie diese in ein bestehendes Displaysystem integriert werden.

7.4.1 Prototyp eines Direct-Lit LCDs mit 180 LEDs

Bei dem Prototyp des Direct-Lit LCDs handelt es sich um ein auf Messungen basierendes Modell eines Direct-Lit TVs mit einer marktüblichen Zahl von 180 LEDs. Die Auflösung des Displays ist Full-HD.

Die für diesen Prototyp verwendete Leuchtdiode wurde mit der digitalen Spiegelreflexkamera Nikon D70 [Nik 10] an einem realen Display vermessen (siehe linke Seite der Abbildung 7-9). Um die fertigungsbedingte Streuung der LEDs zu berücksichtigen, wurde aus den Helligkeitsverteilungen von 10 identischen Dioden (siehe rechte Seite der Abbildung 7-9) ein Mittelwert gebildet.

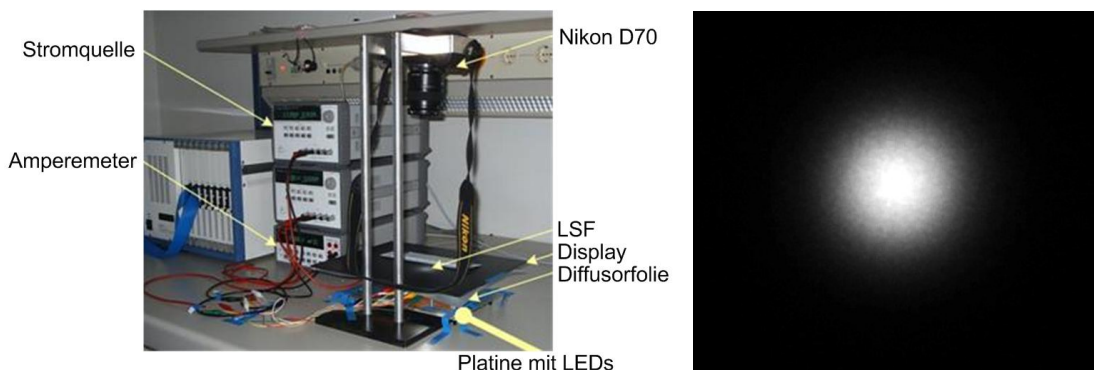


Abbildung 7-9: links: Aufbau zur Messung der Helligkeitsverteilung einer Leuchtdiode; rechts: gemessene Helligkeitsverteilung einer Leuchtdiode

7.4.1.1 LSFs und Einflussmatrix

Die LSF für eine LED (siehe linke Seite der Abbildung 7-10) ergibt sich durch Überlagerung der 12 einzelnen Dioden. Diese LSF wird für alle der 180 LEDs übernommen, unterscheidet sich also nur durch den Ort der Platzierung. Die Einflussmatrix ist dünn besetzt, da die LEDs einen räumlich begrenzten Einfluss auf die Pixel haben. Auf der rechten Seite der Abbildung 7-10 ist die kondensierte Einflussmatrix zu sehen. Es wurden jeweils (20·15) Pixel kondensiert.

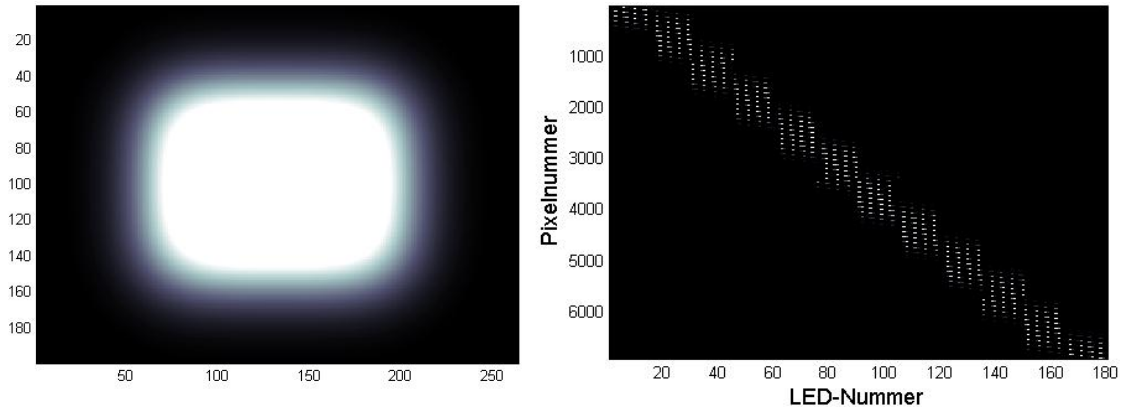


Abbildung 7-10: Direct-Lit Prototyp mit 180 LEDs – links: LSF (vergrößert); rechts: kondensierte Einflussmatrix (20·15) Pixel

7.4.1.2 Adaption der Local Dimming Hardware

Condenser

Mit Hilfe der ermittelten LSF wird analog zum Beispiel aus Abbildung 5-5 die Zellgröße für die Kondensierung bestimmt. Anhand der Einflussmatrix des Backlights (siehe rechte Seite der Abbildung 7-10) lässt sich erkennen, dass eine LED nur auf wenige Pixel des Displays einen maßgeblichen Einfluss hat. Dies hat eine direkte Auswirkung auf die maximale Zellgröße, da die LSF innerhalb weniger Pixel sehr viel Helligkeit hinzugewinnt bzw. verliert. Eine hohe Zellgröße würde sich, bedingt durch die hohe Anzahl von LEDs und die daraus resultierende hohe Granularität des Backlights, negativ auf die Bildqualität auswirken. So beläuft sich die festgelegte Zellgröße auf (20·15) Pixel, wodurch sich die Anzahl der zu prozessierenden Pixel auf 6912 Pixel reduziert.

Es sind drei verschiedene Condenser-Modi implementiert. Neben dem clippingfreien Modus ist ein Maximum-Power-Saving-Modus und ein automatischer Modus mit vier Schwellwerten implementiert.

SSC

Aufgrund der hohen Zahl an LEDs und der Beschaffenheit der Einflussmatrix wird die in Abschnitt 4.6.1 beschriebene lokale Optimierung eingesetzt. Der Algorithmus selbst kommt, dank der dünn besetzten Einflussmatrix, ohne die IP aus, besteht also nur aus LB und FDC. Bei der Prozessierung der FDC wird eine Parallelisierung zweiten Grades verwendet (siehe Abschnitt 4.6.3).

Postprozessor

Der Postprozessor zur Berechnung der TFT-Werte wird mit dem in Abschnitt 6.2 beschriebenen Verfahren implementiert. Die Umsetzung von TFT-Wert und Helligkeit ist linear, die Größe des zu interpolierenden Gebietes beträgt $(16 \cdot 16)$ Pixel.

Verwendete Speicher

Der hier vorgestellte Prototyp verwendet die in Tabelle 7-2 aufgelisteten Speicher.

| Speicher | Größe | Busbreite |
|---------------|-----------|-----------|
| GD-RAM | 6,22 MB | 24bit |
| TFT-RAM | 6,22 MB | 24bit |
| Condenser-RAM | 6,912 KB | 16bit |
| LSF-RAM (SSC) | 34,560 KB | 64bit |
| LSF-RAM (TFT) | 41,745 KB | 64bit |

Tabelle 7-2: Verwendete Speicher des Direct-Lit TVs

7.4.1.3 Ergebnisse der Implementierung

Im Gegensatz zu den beiden noch folgenden Prototypen wurde hier keine FPGA-Implementierung vorgenommen, da es sich um ein Modell eines Displays handelt. Der in Tabelle 7-3 beschriebene Aufwand an Hardware bezieht sich auf eine Hardwired-Implementierung für ein ASIC in Verilog [Hop 06].

| Prozessorfrequenz | # Gatter | Bilder/ Sekunde | Bildgröße |
|-------------------|----------|-----------------|-----------------------------------|
| 150 MHz | 308521 | 50 | $(1920 \cdot 1080 \cdot 3)$ Pixel |

Tabelle 7-3: Ergebnisse der Implementierung des Direct-Lit TVs

Bei einer Prozessorfrequenz von 150 MHz werden für 50 Frames pro Sekunde und einer Bildgröße von Full-HD 308521 Gatter benötigt. Die dafür benötigte Chipfläche beträgt nur einen Bruchteil derer, die vom GD-RAM benötigt wird.

7.4.2 Prototyp eines Edge-Lit LCDs mit 6 LEDs

Bei dem nun vorgestellten Prototyp handelt es sich um ein kommerzielles Edge-Lit LCD von Samsung (Samsung UE32B6000 [Sa 09]) mit einer Full-HD-Auflösung. Die einzige Modifikation, die an dem Gerät vorgenommen wurde, ist, dass die 6 LED-Stränge nach außen geführt und von einer eigenen Treiberplatine mit Strom versorgt werden können. Die Platine wird von einem FPGA-Board angesteuert, auf dem die PWM-Werte für die LEDs erzeugt werden. Auf der rechten Seite der Abbildung 7-11 ist das Display nach der oben beschriebenen Modifikation zu sehen. Die linke Seite zeigt eine Fotografie des Displays im laufenden Betrieb.

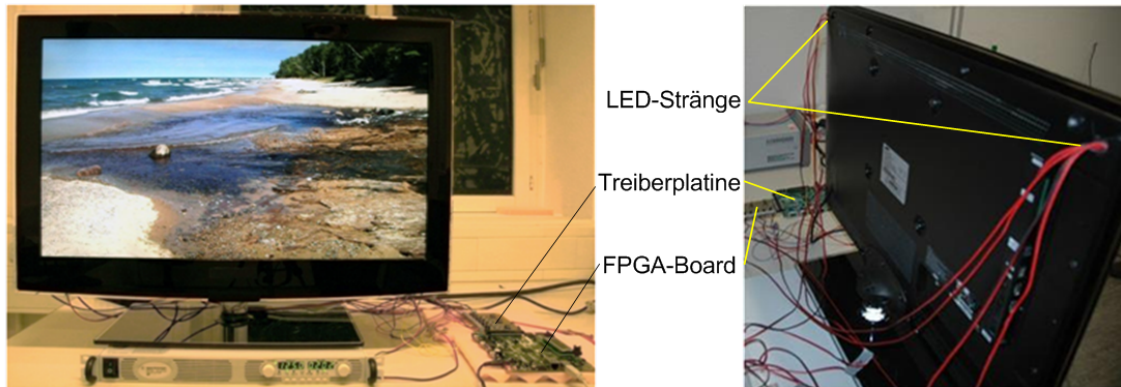


Abbildung 7-11: Fotografien des Edge-Lit-Prototyps mit nach außen geführten LEDs – links: Vorderansicht bei laufendem Betrieb; rechts: Rückansicht

7.4.2.1 LSFs und Einflussmatrix

Die LSFs der sechs LEDs wurden messtechnisch mit einer Nikon D70 ermittelt (siehe Abbildung 7-12).

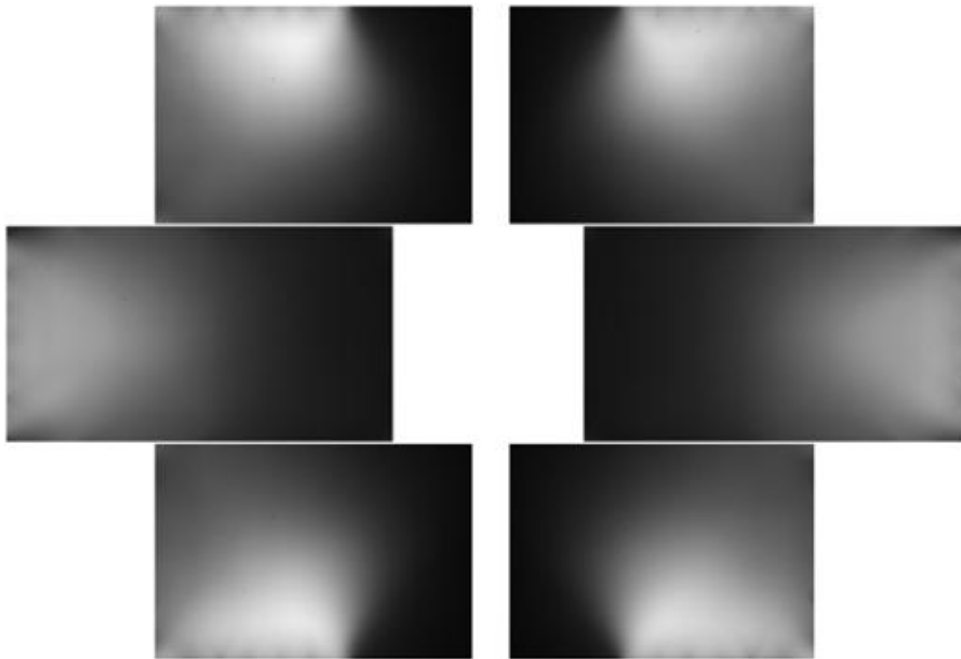


Abbildung 7-12: LSFs des Edge-Lit-Prototyps

Es fällt auf, dass eine LED in etwa die Hälfte aller Pixel merklich beeinflusst. Aus diesem Grund ist die Einflussmatrix dieses Displays (siehe Abbildung 7-13 in bereits kondensierter Form) dichter besetzt als die des Prototyps aus Abschnitt 7.4.1.

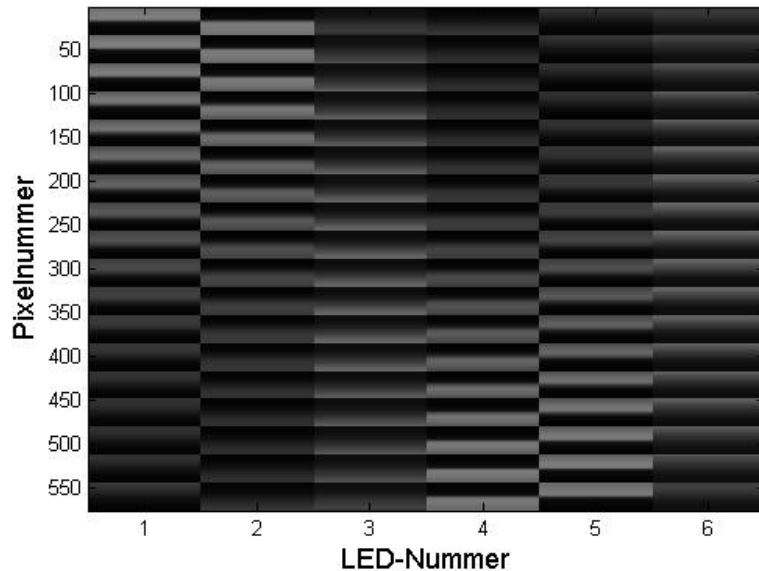


Abbildung 7-13: Kondensierte Einflussmatrix (60·60) Pixel des Edge-Lit-Prototyps

7.4.2.2 Adaption der Local Dimming Hardware

Condenser

Die Beschaffenheit der Einflussmatrix lässt eine hohe Zellgröße von (60·60) Pixeln zu. Das führt zu einer Reduktion der Komplexität auf 0,27%, wodurch die Anzahl der zu prozessierenden Pixel auf 576 reduziert wird.

Es werden drei verschiedene Condenser-Modi implementiert. Neben dem clippingfreien Modus ist ein Maximum-Power-Saving-Modus und ein automatischer Modus mit vier Schwellwerten implementiert.

SSC

Der Algorithmus besteht, neben den LB und FDC, ebenfalls aus der IP. Diese ist festgelegt auf neun Iterationen. Es wird die in Abschnitt 4.6.2 gezeigte globale Optimierung gewählt. Eine Parallelisierung der Berechnung erfolgt nicht.

Postprozessor

Der Postprozessor interpoliert zwischen Stützstellen mit dem Abstand von (64·64) Pixel.

Verwendete Speicher

Der hier vorgestellte Prototyp verwendet die in Tabelle 7-4 aufgelisteten Speicher.

| Speicher | Größe | Speichertyp | Busbreite |
|---------------|----------|-------------|-----------|
| GD-RAM | 6,22 MB | DDR2-SDRAM | 32bit |
| Condenser-RAM | 4,61 kB | SRAM | 8bit |
| LSF-RAM (SSC) | 34,56 kB | SRAM | 64bit |
| LSF-RAM (TFT) | 39,06 kB | SRAM | 64bit |

Tabelle 7-4: Verwendete Speicher des Edge-Lit TVs

Neben dem obligatorischen GD-RAM wird hier nur der FPGA-interne SRAM benötigt.

7.4.2.3 Ergebnisse der Implementierung

Der Edge-Lit-Prototyp mit 6 LEDs wurde mit Verilog und VHDL [Hei 02] auf ein Altera Cyclone II FPGA [Alt 10] implementiert. Tabelle 7-5 zeigt das Ergebnis der Implementierung.

| | # Gatter | Bilder/ Sekunde | Bildgröße |
|---------------------|-----------------------|--------------------|------------------------|
| Logikblöcke | 10830 / 18752 (58%) | 50 | (1920·1080·3) Pixel |
| 9bit-Multiplizierer | 52 / 52 (100%) | | |
| SRAM | 143960 / 239616 (60%) | | |

Tabelle 7-5: Ergebnisse der Implementierung des Edge-Lit TVs

Die vom FPGA bereitgestellten Ressourcen genügen zur Implementierung der Local Dimming Hardware. Es wird insgesamt nur ein kleiner Teil der Hardware benötigt.

7.4.2.4 Integration der Local-Dimming-Prozessierung

Abbildung 7-14 zeigt das Timing-Diagramm der Edge-Lit-Implementierung.

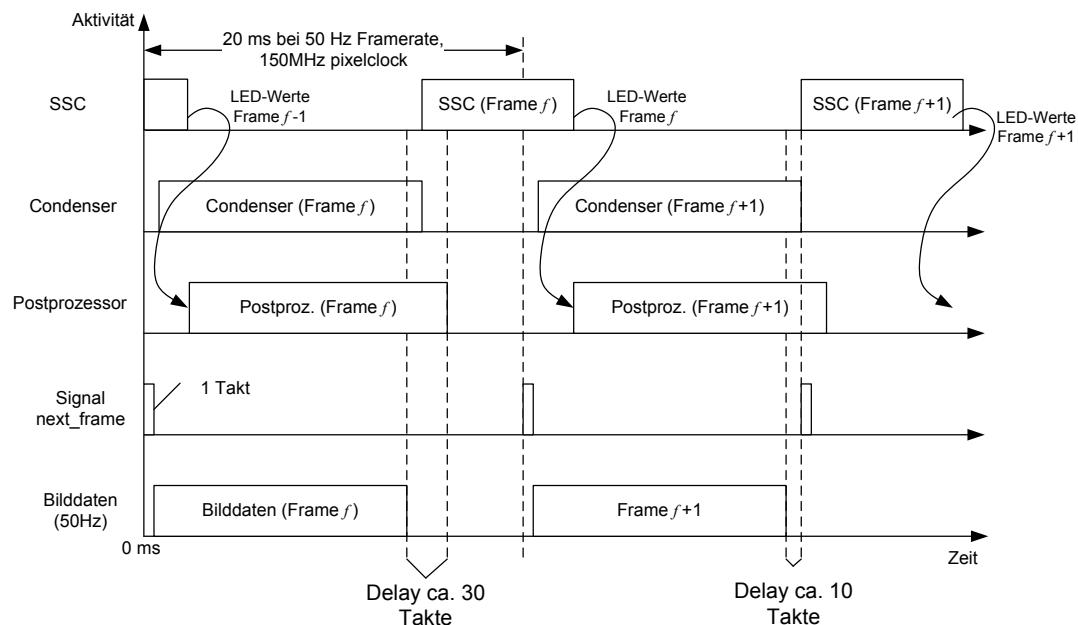


Abbildung 7-14: Timing-Diagramm der Edge-Lit-Implementierung

Über das Signal „next_frame“ wird der Local Dimming Hardware der Übergang zu einem neuen Bild mitgeteilt. Von diesem Zeitpunkt an werden mit jedem neuen Takt die Daten für einen Pixel übermittelt (3·10bit). Die Bilddaten von Frame (f) werden parallel kondensiert und über die erreichte Helligkeit mit den LED-Werten von Frame ($f-1$) postprozessiert. Sobald die Kondensierung beendet ist, startet die Berechnung der LED-Werte. Die Verzögerung zwischen Eingang der Bilddaten und Ausgabe der Bilddaten (siehe auch Abbildung 7-6) beläuft sich auf

circa 30 Takte. Die Berechnung der LED-Werte ist beendet, bevor ein neues Bild eingelesen wird.

7.4.3 Prototyp einer Signalanzeige mit 72 LEDs

Beim letzten Prototyp handelt es sich um eine Signalanzeige mit 72 LEDs, wie sie an Bahnsteigen zum Einsatz kommt. Abbildung 7-15 zeigt eine Fotografie der Signalanzeige. Das Display ist zusammengesetzt aus 6 kleinen Displays. Die Gesamtauflösung beträgt (576·104) Pixel.

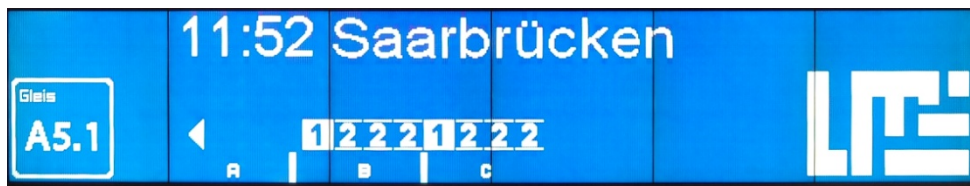


Abbildung 7-15: Fotografie der Signalanzeige

Im Gegensatz zu den beiden anderen Prototypen handelt es sich hier um ein monochromes Display. Das bedeutet, dass eine LC-Zelle nur ein- bzw. ausgeschaltet sein kann. Im eingeschalteten Zustand wird die komplette Helligkeit durchgelassen, der Pixel erscheint weiß. Im ausgeschalteten Zustand erscheint der Pixel jedoch nicht wie üblich schwarz, sondern blau. Dies beruht auf der Dispersion der Flüssigkristalle, welchen ein wellenlängenabhängiger Brechungsindex zugeordnet ist. Das Blau leuchtet umso stärker, je mehr Helligkeit erzeugt wird. Abhängig von der Umgebungshelligkeit muss diese Tatsache bei der Prozessierung berücksichtigt werden. Dadurch kann der minimal erlaubte LED-Wert größer als null sein.

7.4.3.1 LSFs und Einflussmatrix

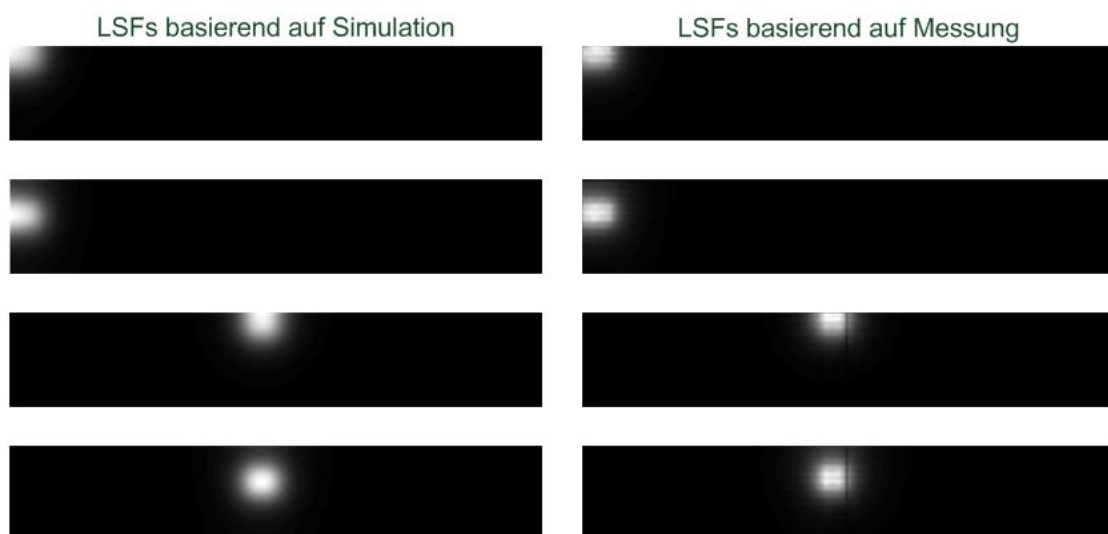


Abbildung 7-16: Die vier charakteristischen LSFs der Signalanzeige - links: physikalische Simulation mit SPEOS; rechts: Messung am realen Display

Abbildung 7-16 zeigt die vier charakteristischen der insgesamt 72 LSFs. Die Daten auf der linken Seite der Abbildung entstammen einer physikalischen Simulation mit der Software SPEOS [SPE 10]. Diese wurden messtechnisch mit der Nikon D70 verifiziert (siehe rechte Seite der Abbildung 7-16). Mit Hilfe der hier gezeigten vier LSFs können die restlichen LSFs durch Verschieben bzw. Spiegeln erzeugt werden.

Die daraus resultierende, kondensierte Einflussmatrix der Signalanzeige ist in Abbildung 7-17 zu sehen.

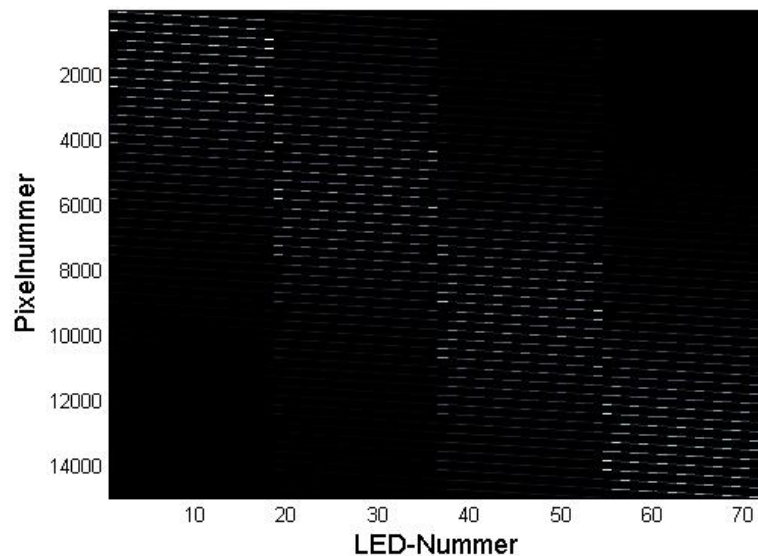


Abbildung 7-17: Einflussmatrix der Signalanzeige, kondensiert sind jeweils (2·2) Pixel

Diese ist ähnlich dicht besetzt wie die Einflussmatrix des Edge-Lit TVs. Allerdings werden hier zwölfmal mehr LEDs verwendet.

7.4.3.2 Adaption der Local Dimming Hardware

Condenser

Die Signalanzeige verfügt über eine weit geringere Auflösung als die anderen beiden Prototypen. Eine Kondensierung zur Reduktion der Komplexität ist daher nicht zwingend notwendig.

Clipping kann aufgrund des Einsatzbereiches des Displays nur in sehr geringem Maße zugelassen werden. Um Speicherplatz zu sparen, werden jeweils (2·2) Pixel im clippingfreien Modus zu einem Pixel kondensiert. In jedem kondensierten Pixel steht folglich der maximale Wert der kondensierten Zelle.

Da sich der Leistungsverbrauch dieser Anzeige auf 500 Watt beläuft, lohnt es sich, diesen noch weiter zu verbessern. Aus diesem Grund wurde eine Methode entwickelt, die einen Anteil Δ_b vom eigentlichen Grauton des Bildes subtrahiert. Als Resultat erhält man ein Bild, bei dem die Übergänge zwischen den Kanten abgerundet sind. Die Auswirkungen dieser Vorgehensweise

auf die Leistungersparnis und die visuelle Qualität werden in den Abschnitten 8.2.1.3 und 8.3.1.3 untersucht.

SSC

Bedingt durch die Beschaffenheit der Einflussmatrix muss auch hier, neben den LB und FDC, die IP verwendet werden. Diese besteht aus vier Iterationen. Es wird die in Abschnitt 4.6.2 gezeigte globale Optimierung verwendet. Allerdings werden hier aus Gründen der Speicherreduktion nur die 32 einflussreichsten der insgesamt 72 LEDs pro Pixel bei der Berechnung der LED-Werte berücksichtigt. Das Vernachlässigen der Einflüsse der weiter entfernten LEDs hat nur eine sehr geringe Auswirkung auf die Leistungersparnis. Diese beträgt nur wenige Promille.

Postprozessor

Da es sich bei der Signalanzeige um ein monochromes Display handelt, ist keine Postprozessierung erforderlich. Die Bilddaten werden unverändert an das LCD-Modul weitergeleitet.

Verwendete Speicher:

Der hier vorgestellte Prototyp benötigt die in Tabelle 7-6 aufgelisteten Speicher.

| Speicher | Größe | Speichertyp | Busbreite |
|---------------|----------|-------------|-----------|
| GD-RAM | 59,9 kB | DDR2-SDRAM | 64bit |
| Condenser-RAM | 14,63 kB | SRAM | 8bit |
| LSF-RAM (SSC) | 1,02 MB | DDR2-SDRAM | 64bit |

Tabelle 7-6: Verwendete Speicher der Signalanzeige

Beim GD-RAM handelt es sich um einen externen DDR2-SDRAM, in welchen ebenfalls der LSF-RAM für SSC integriert wird. Der Condenser-RAM ist in den SRAM des FPGAs integriert, wodurch dieser zu 12% ausgelastet ist.

7.4.3.3 Ergebnisse der Implementierung

Die Signalanzeige wurde mit der VHDL in ein Altera Cyclone III FPGA [Alt 10] implementiert. Tabelle 7-7 zeigt die Ergebnisse der Implementierung.

| | # Gatter/ Farbe | Bilder/ Sekunde | Bildgröße |
|---------------------|-------------------|-----------------|--------------------|
| Logikblöcke | 3530 / 39600 (9%) | 25 | (576·104) Pixel |
| 9bit-Multiplizierer | 4 / 252 (2%) | | |
| SRAM | 1167 (12%) | | |

Tabelle 7-7: Ergebnisse der Implementierung der Signalanzeige

Man erkennt, dass nur ein kleiner Teil der FPGA Hardware für die Implementierung benötigt wird. So werden nur 9% der Gatter und 2% der Multiplizierer verwendet.

7.4.3.4 Integration der Local-Dimming-Prozessierung

Durch den geringen Bedarf an Hardware kann der Local-Dimming-Prozessor in die bereits bestehenden Ressourcen integriert werden. Abbildung 7-18 zeigt das Blockdiagramm der verwendeten Hardware.

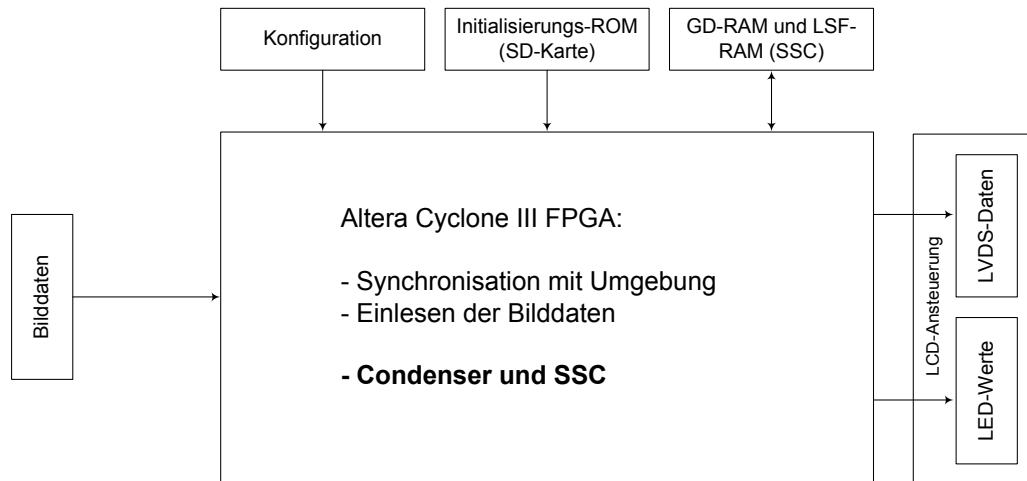


Abbildung 7-18: Blockdiagramm der Hardware und der Schnittstellen zur Umgebung

In einem ersten Schritt werden parallel die Pixelwerte in den GD-RAM und die kondensierten Werte in den SRAM geschrieben. Bei der anschließenden Berechnung der LED-Werte muss auf den LSF-RAM zugegriffen werden, jedoch kann man dabei Zugriffskonflikte ausschließen, da die Berechnung der LED-Werte bis zum nächsten Beschreiben des GD-RAMs abgeschlossen ist. Die LED-Werte und die Pixelwerte werden dann an die LCD-Ansteuerung weitergegeben.

7.5 Fazit

In diesem Kapitel wurde gezeigt, wie der Local-Dimming-Prozessor, bestehend aus Condenser, SSC und Postprozessor, in Hardware implementiert werden kann. Die Umsetzung in Hardware wurde dann an drei unterschiedlichen Local-Dimming-Prototypen vollzogen.

Anhand der Implementierungsergebnisse der drei Prototypen wird deutlich, dass der benötigte Aufwand an Hardware derart gering ist, dass der Local-Dimming-Prozessor auf einem kostengünstigen FPGA implementiert werden kann. Der größte Anteil (ca. 50%) der logischen Elemente wird dabei vom Postprozessor benötigt, obwohl dieser in einer platzsparenden Pipelinearchitektur realisiert wird. Daran erkennt man, wie effizient die in dieser Arbeit entworfenen Komponenten, Condenser und SSC-Prozessor, sind.

Der zusätzliche Speicher für das kondensierte Bild und die LSFs beträgt nur einen Bruchteil des Speichers, der für ein Bild benötigt wird.

Eine bereits bestehende Hardware wird durch das Hinzufügen des Local-Dimming-Prozessors in ihrer Funktion nicht beeinflusst. Der Bilddatenstrom und die Synchronisationssignale zwischen Videoprozessor und LCD-Modul werden lediglich für wenige Takte gepuffert und

gemeinsam mit den errechneten LED-Werten an den Timing-Controller im LCD-Modul übergeben.

8 Statistische und visuelle Ergebnisse

8.1 Einleitung

In Abschnitt 7.4 wurden drei in Hardware implementierte Prototypen vorgestellt. Diese werden zunächst bezüglich ihrer statistischen Ergebnisse untersucht. Dabei wird ein Augenmerk auf die Leistungersparnis und die relative Abweichung von der optimalen Lösung gelegt. Im Anschluss daran werden die visuellen Ergebnisse der drei Prototypen präsentiert und ausgewertet.

8.2 Statistische Ergebnisse

8.2.1 Leistungersparnis der Prototypen

Die Leistungersparnis der ersten beiden Prototypen (Direct-Lit TV und Edge-Lit TV) wird am Standard IEC 62087 [IEC 09] untersucht und ausgewertet. Bei diesem Standard handelt es sich um ein etwa 10-minütiges Testvideo, dessen Inhalt so ausgewählt wurde, dass es einen Durchschnitt des weltweit ausgestrahlten Fernsehprogramms darstellt. Abbildung 8-1 zeigt einen Auszug aus dem 18578 Frames umfassenden Testvideo.

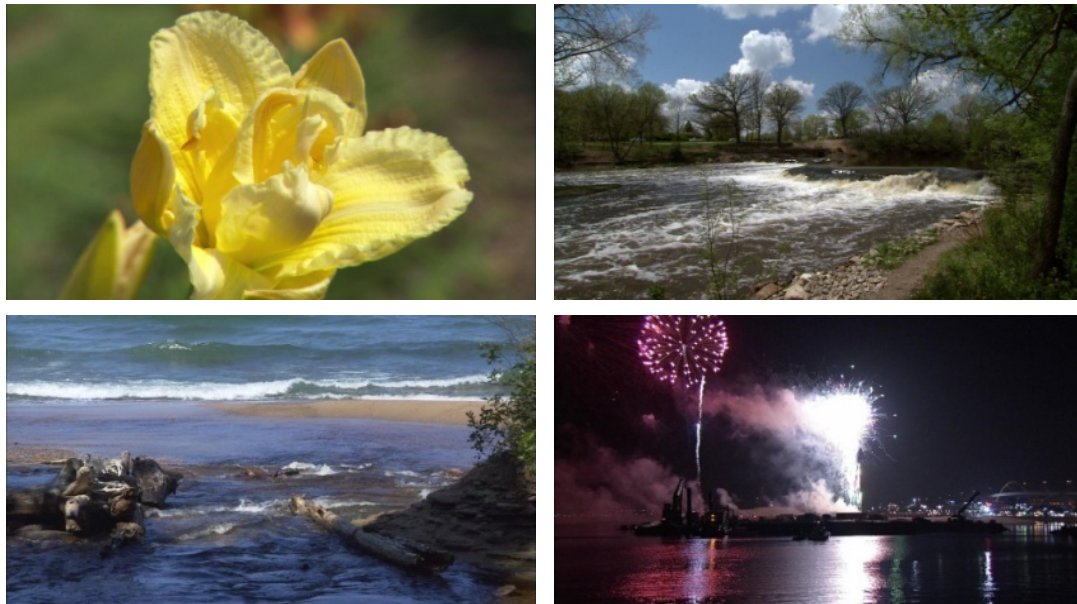


Abbildung 8-1: Auszug aus dem Testvideo für die ersten beiden Prototypen

Für den dritten Prototyp, die Direct-Lit-Signalanzeige, wird ein Test-Set von sieben Bildern verwendet (siehe Abbildung 8-2).



Abbildung 8-2: Test-Set für die Direct-Lit-Signalanzeige

Die erreichte Leistungersparnis bezieht sich auf die Lösung des OGD (siehe Abschnitt 4.5.2). Diese entspricht in der Regel der Summe der LED-Werte bei einem ungedimmten Backlight. Die eingesparte Leistung ergibt sich aus Formel 8.1:

$$\text{Leistungersparnis [\%]} = \left(1 - \frac{\sum_{\ell=1}^L x(\ell) \Big|_{SSC}}{\sum_{\ell=1}^L x(\ell) \Big|_{OGD}} \right) \cdot 100 \quad 8.1$$

8.2.1.1 Direct-Lit TV

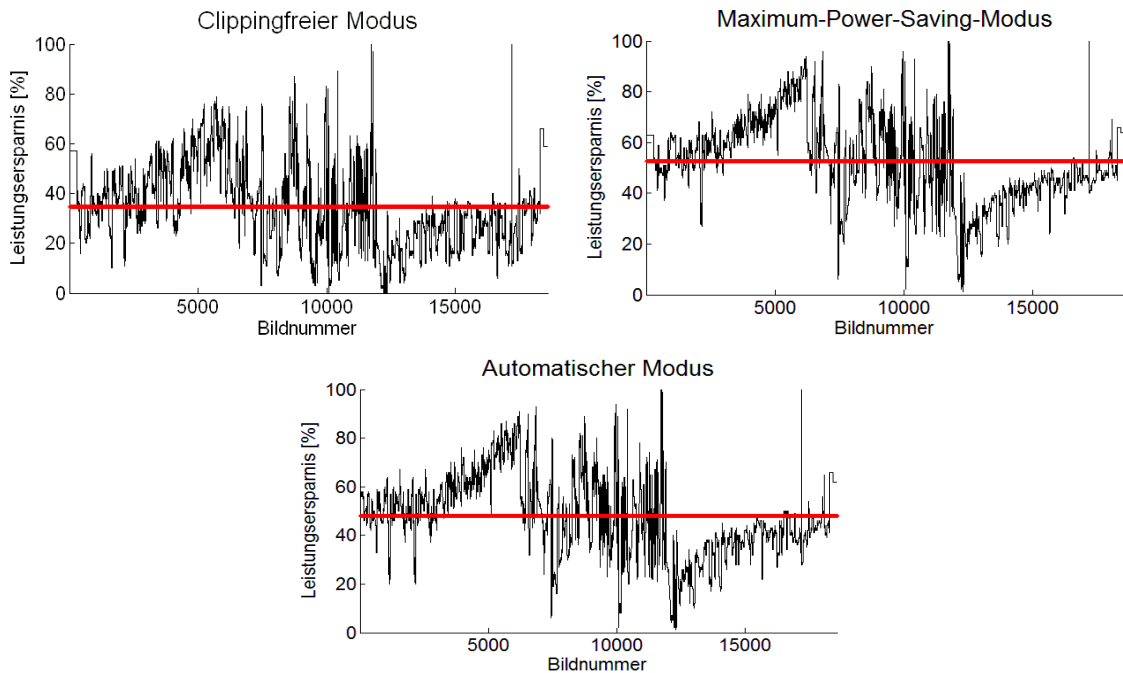


Abbildung 8-3: Leistungersparnis des Direct-Lit TVs (von oben links nach unten: clippingfreier Modus, Maximum-Power-Saving-Modus und automatischer Modus)

Die statistischen Ergebnisse der drei Condenser-Modi für das Direct-Lit TV sind in Form eines qualitativen Vergleiches in Abbildung 8-3 zu sehen.

Im Allgemeinen gilt, dass bei hellen Bildern wenig Energie und bei dunklen Bildern viel Energie eingespart werden kann. Im clippingfreien Modus wird eine Leistungersparnis von durchschnittlich 34% erzielt. Im automatischen Modus beträgt die Ersparnis 48% und im Maximum-Power-Saving-Modus 52% im Durchschnitt. Durch die hohe Anzahl von LEDs ist es möglich, das Backlight sehr lokal an den Bildinhalt zu adaptieren. Dadurch lässt sich die hohe Einsparung im clippingfreien Modus erklären.

8.2.1.2 Edge-Lit TV

Abbildung 8-4 zeigt die Ergebnisse der Leistungersparnis beim Edge-Lit TV. Auch hier wurden drei verschiedene Condenser-Modi simuliert.

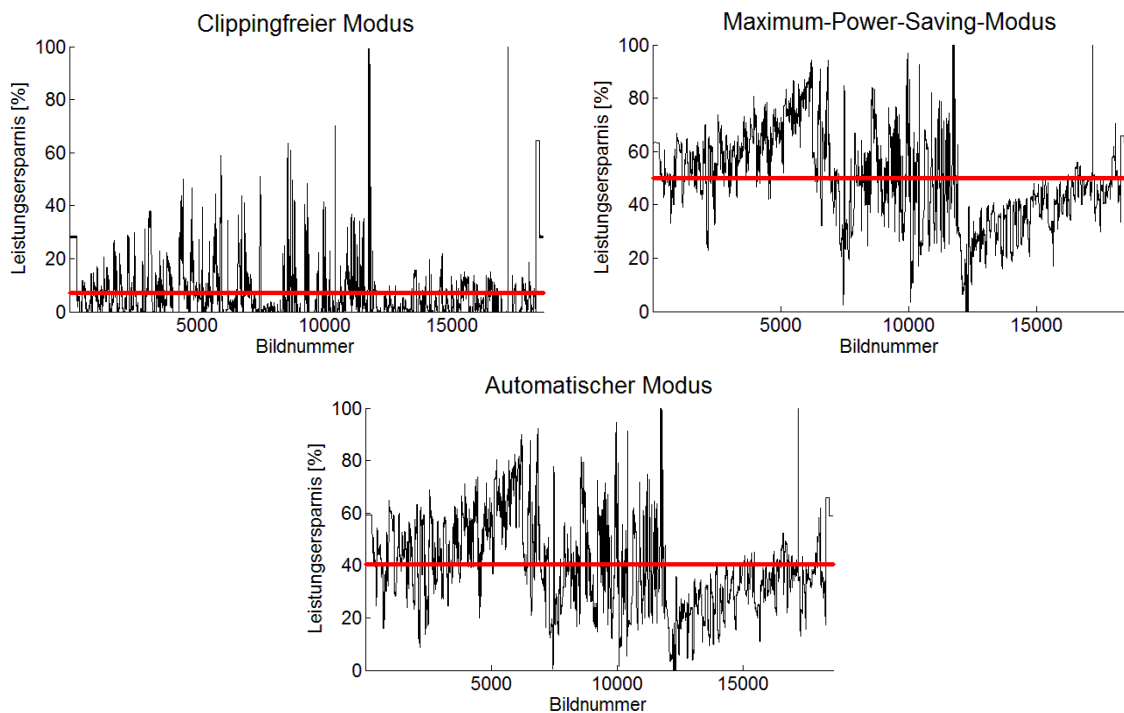


Abbildung 8-4: Leistungersparnis des Edge-Lit TVs (von oben links nach unten: clippingfreier Modus, Maximum-Power-Saving-Modus und automatischer Modus)

Aufgrund der geringen Granularität und der Beschaffenheit der LSFs des Backlights kann im clippingfreien Modus mit 7% nur eine sehr geringe Leistungersparnis erreicht werden. Hier bewirkt bereits ein heller Pixel in der Mitte des Bildes, dass alle LEDs auf den maximalen Wert eingestellt werden müssen. Erlaubt man jedoch Clipping, erhöht sich die Leistungersparnis drastisch. Im automatischen Modus beträgt diese schon 40%, im Maximum-Power-Saving-Modus sogar 51%.

8.2.1.3 Direct-Lit-Signalanzeige

Die erreichte Leistungersparnis der Direct-Lit-Signalanzeige für das Test-Set aus Abbildung 8-2 ist in Abbildung 8-5 zu sehen.

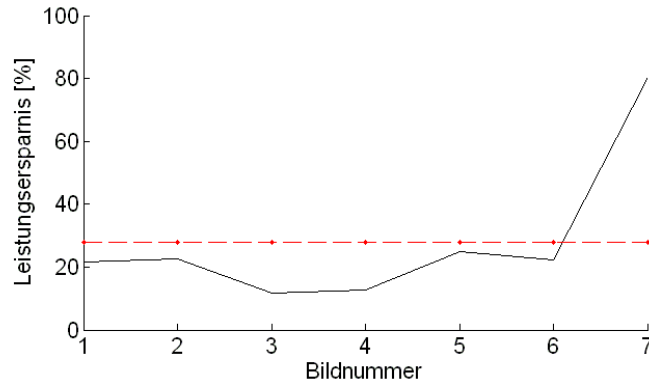


Abbildung 8-5: Leistungersparnis der Direct-Lit-Signalanzeige

Aufgrund der Forderung nach möglichst hohen Kontrastwerten wird die Signalanzeige ausschließlich im clippingfreien Modus betrieben. Die durchschnittliche Leistungersparnis beträgt hier 27,4%.

Diese kann durch Variation des in Abschnitt 7.4.3.2 besprochenen Δ_b noch weiter erhöht werden (siehe Tabelle 8-1).

| Δ_b | 1 | 2 | 3 | 4 | 5 | 6 |
|-------------------|-------|-------|-----|-------|-------|-------|
| Leistungersparnis | 27,9% | 28,4% | 29% | 29,5% | 30,1% | 30,6% |

Tabelle 8-1: Leistungersparnis der Direct-Lit-Signalanzeige bei Variation von Δ_b

Es handelt sich hierbei um eine tatsächliche Erhöhung der Leistungseinsparung. So ergibt sich beispielsweise im Vergleich zu $\Delta_b = 0$ eine um 3,2% höhere Leistungersparnis bei $\Delta_b = 6$, obwohl die Helligkeit eines Pixels maximal um 2,4% (6/255) herabgesetzt wird.

In Abschnitt 8.3.1.3 wird gezeigt, dass diese Methode nur geringe, nicht wahrnehmbare Auswirkungen auf die visuelle Qualität der prozessierten Bilder hat.

8.2.2 Qualität der Ergebnisse

Die in Abschnitt 3.3.2 vorgestellte Linear Programming Software CPLEX wird zur Berechnung der optimalen Lösung verwendet, welche als Referenz für die mit SSC erzielten Ergebnisse dient. Da CPLEX für ein Full-HD Bild mit 180 LEDs etwa 20 Minuten Berechnungszeit auf einem Rechnercluster benötigt, würde eine Simulation aller 18578 Testbilder mehrere Monate dauern. Aus diesem Grund wird für alle Prototypen ein auf 100 Testbilder reduziertes Test-Set simuliert. Abbildung 8-6 zeigt einen Auszug des Test-Sets.



Abbildung 8-6: Auszug aus dem Test-Set für die ersten beiden Prototypen

Für den dritten Prototyp werden die Bilder des Test-Sets auf die hier benötigte Auflösung von (576·104) Pixel skaliert. Darüber hinaus wird jeweils nur der maximale der drei Subpixel verwendet (siehe Abbildung 8-7).



Abbildung 8-7: Auszug aus dem Test-Set für die Direct-Lit-Signalanzeige

Die relative Abweichung der mit SSC erzielten Ergebnisse von CPLEX wird über die folgende Formel berechnet:

$$\text{relative Abweichung [\%]} = \left(\frac{\sum_{\ell=1}^L x(\ell) \Big|_{SSC}}{\sum_{\ell=1}^L x(\ell) \Big|_{CPLEX}} - 1 \right) \cdot 100 \quad 8.2$$

Für alle drei Prototypen ist gewährleistet, dass die Randbedingungen für CPLEX und SSC jeweils identisch sind. Sowohl CPLEX als auch SSC erhalten jeweils die gleiche Einflussmatrix und die gleichen Bilddaten.

8.2.2.1 Direct-Lit TV

Abbildung 8-8 zeigt die relative Abweichung der Ergebnisse von CPLEX und SSC beim Direct-Lit TV.

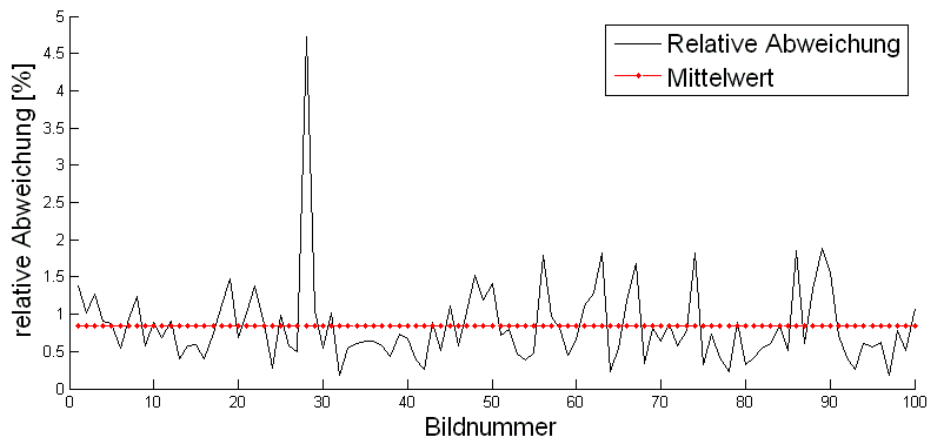


Abbildung 8-8: Relative Abweichung der Ergebnisse von CPLEX und SSC beim Direct-Lit TV

Die relative Abweichung beträgt im Durchschnitt 0,84%. Es gibt nur ein Testbild, bei dem die relative Abweichung über 2% liegt (siehe Abbildung 8-9).



Abbildung 8-9: Bild mit größter relativer Abweichung der Ergebnisse von CPLEX und SSC beim Direct-Lit TV

Bei dieser Ausnahme handelt es sich jedoch um ein sehr dunkles Bild, bei dem 90% der Leistung eingespart werden kann. Die höhere relative Abweichung von 4,7% lässt sich durch die geringe Anzahl der von CPLEX benötigten LED-Werte erklären. So hat eine geringe absolute Abweichung der von SSC ermittelten Summe der LED-Werte eine hohe relative Abweichung von der von CPLEX ermittelten Summe der LED-Werte zur Folge.

8.2.2.2 Edge-Lit TV

Die relative Abweichung der Ergebnisse von CPLEX und SSC beim Edge-Lit TV ist in Abbildung 8-10 zu sehen.

Diese liegt im Mittel bei 1,48% und ist damit höher als die beim Direct-Lit TV. Die höhere Abweichung lässt sich dadurch erklären, dass nur sechs LEDs verwendet werden, wodurch ein

unnötiges Erhöhen einer LED (vergleiche Abschnitt 4.11.1) eine größere Auswirkung auf die Summe der LED-Werte hat.

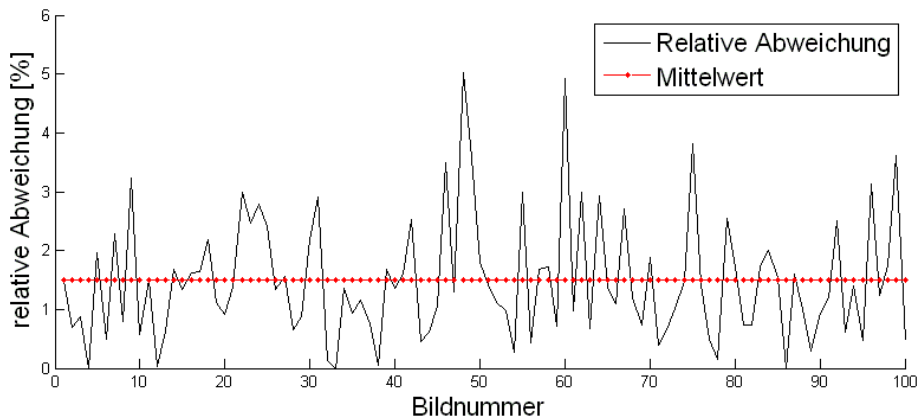


Abbildung 8-10: Relative Abweichung der Ergebnisse von CPLEX und SSC beim Edge-Lit TV

8.2.2.3 Direct-Lit-Signalanzeige

Die relative Abweichung der Ergebnisse von CPLEX und SSC bei der Signalanzeige ist in Abbildung 8-11 zu sehen.

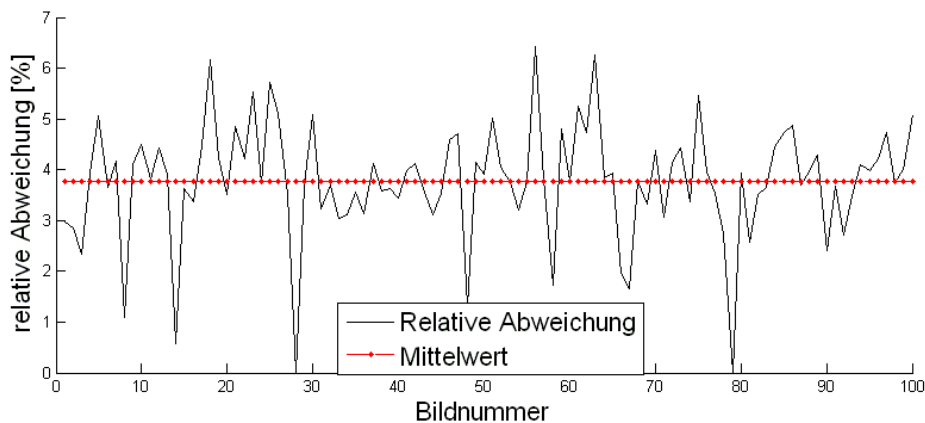


Abbildung 8-11: Relative Abweichung der Ergebnisse von CPLEX und SSC bei der Direct-Lit-Signalanzeige

Mit einer durchschnittlichen relativen Abweichung von 3,7% ist die Qualität der Ergebnisse hier schlechter als die bei den anderen beiden Prototypen. Dies lässt sich durch die in Abschnitt 4.11.1 beschriebenen Gründe erklären:

Die Einflussmatrix der Signalanzeige ist ähnlich dicht besetzt wie die des Edge-Lit Displays, hat jedoch zwölfmal mehr LEDs. Jeder Pixel wird nicht von sechs, sondern mindestens von 44 LEDs beeinflusst. Des Weiteren werden nur vier Iterationen in der IP ausgeführt. Das sukzessive Erhöhen der LEDs erfolgt somit in größeren Schritten, wodurch die Gefahr besteht, dass ein LED-Wert zu schnell zu weit erhöht und somit Leistung unnötig verbraucht wird.

Eine Verbesserung kann hier durch ein Erhöhen der Anzahl von Iterationen erzielt werden. Bei 10 Iterationen beträgt die relative Abweichung von SSC und CPLEX im Mittel 2,7%.

8.2.3 Fazit

Die statistischen Ergebnisse aus diesem Abschnitt sind in Tabelle 8-2 noch einmal zusammengefasst.

| | Direct-Lit TV | Edge-Lit TV | Signalanzeige |
|-----------------------------------|---------------|-------------|---------------|
| Clippingfreier Modus | 34% | 7% | 27% |
| Automatischer Modus | 48% | 40% | nicht |
| Maximum-Power-Saving-Modus | 52% | 51% | implementiert |
| Relative Abweichung | 0,84% | 1,48% | 3,7% |

Tabelle 8-2: Vergleich der statistischen Ergebnisse der drei Prototypen

Die Leistungersparnis variiert bei den ersten beiden Prototypen je nach Condenser-Modus zwischen 34% und 52% bzw. zwischen 7% und 51%. Durch eine aggressivere Vorgehensweise beim Maximum-Power-Saving-Modus könnte die Leistungseinsparung weiter erhöht werden. Jedoch wird im Rahmen dieser Arbeit zu Gunsten der Bildqualität auf diese Erhöhung verzichtet. Die Leistungersparnis der Signalanzeige beläuft sich auf 27%, kann jedoch durch die Verwendung des Δ_B weiter erhöht werden.

Die durchschnittliche relative Abweichung von der optimalen Lösung liegt, je nach Prototyp, bei 0,84%, 1,48% oder 3,7%, wodurch eine nahezu optimale Leistungersparnis erreicht wird. Natürlich gibt es bei der Abweichung von der optimalen Lösung, je nach aktuell prozessiertem Bild, immer auch vereinzelte Ausrutscher nach oben und unten.

8.3 Visuelle Ergebnisse

8.3.1 Unbewegte Bilder

Im vorangegangenen Abschnitt wurde gezeigt, welches Mehr an Leistung durch das Zulassen von Clipping eingespart werden kann. Ein Zulassen von Clipping ist jedoch dann nicht mehr sinnvoll, wenn das prozessierte Bild dadurch in seiner visuellen Qualität beeinträchtigt wird. Das menschliche Auge allein entscheidet, wann sich Clipping negativ auf die visuelle Qualität eines Bildes auswirkt. In diesem Abschnitt werden die Ergebnisse der Local-Dimming-Prozessierung auf ihre visuelle Qualität hin überprüft.

Zuerst erfolgt eine Auswertung der visuellen Qualität unbewegter Bilder bei allen der in Abschnitt 7.4 gezeigten Prototypen.

Bei Videoanwendungen liegen die LED-Werte, wie in Abschnitt 7.3.3 gezeigt, im Allgemeinen erst für das nachfolgende Bild vor. Welche Auswirkungen dies auf die visuelle Qualität der prozessierten Bilder hat, wird in Abschnitt 8.3.2 gezeigt.

8.3.1.1 Direct-Lit TV

Clippingfreier Modus

In Abschnitt 5.4.1.1 wird erwähnt, dass selbst im clippingfreien Modus durch die Kondensierung der LSFs im prozessierten Bild Clipping auftreten kann, welches allerdings für das menschliche Auge nicht wahrnehmbar ist. Diese Aussage soll nun anhand eines Beispielbildes verifiziert werden. Abbildung 8-12 zeigt den Vergleich zwischen Originalbild und dem Ergebnis der Prozessierung im clippingfreien Modus.

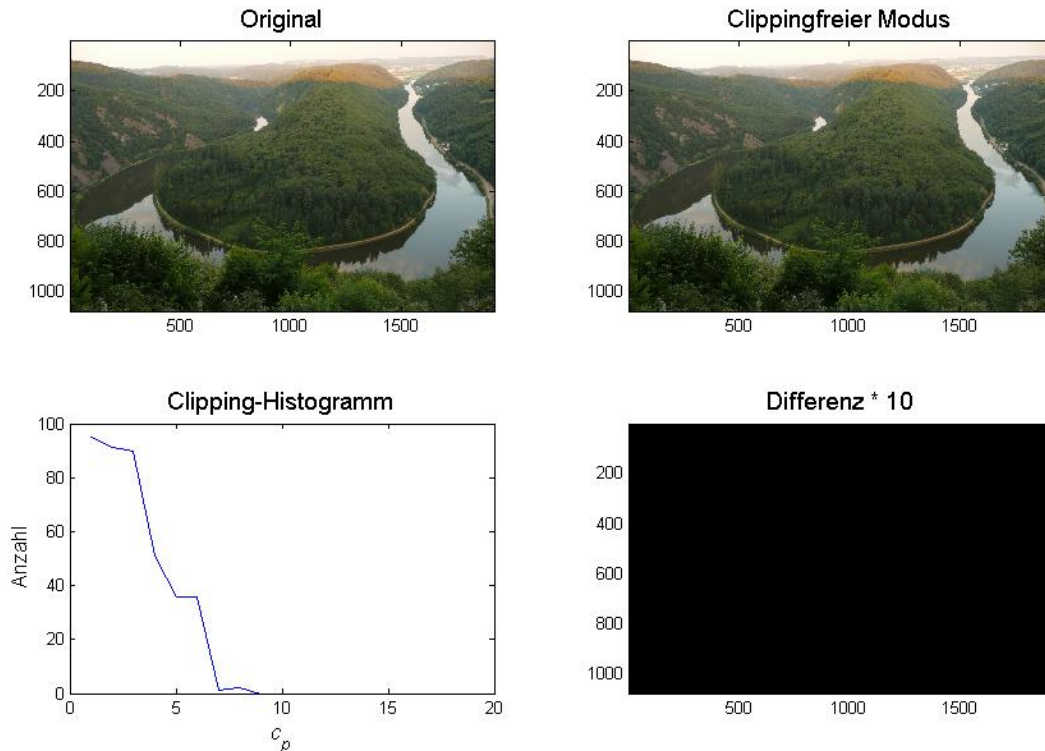


Abbildung 8-12: oben: Vergleich zwischen Originalbild (links) und Ergebnis der Prozessierung im clippingfreien Modus (rechts); unten: Clipping-Histogramm (links) und Differenz zwischen Original und prozessiertem Bild um Faktor 10 verstärkt (rechts)

Mit dem bloßen Auge ist zwischen dem Originalbild und dem im clippingfreien Modus prozessierten Bild kein Unterschied zu erkennen. Die erreichte Bildqualität in diesem Modus ist perfekt. Das Clipping-Histogramm zeigt, dass nur sehr wenige der über 6 Millionen Subpixel des Bildes von Clipping betroffen sind. Dies liegt, wie bereits in Abschnitt 5.3 erwähnt, an der Mittelwertbildung bei der Kondensierung der LSFs. Würde man den minimalen LSF-Wert der Zelle verwenden, wären die Ergebnisse frei von Clipping. Die clippingbehafteten Subpixel treten so über das Bild verteilt auf, dass eine Verschlechterung der visuellen Qualität des Bildes ausgeschlossen werden kann. Die Leistungersparnis für das oben gezeigte Bild beträgt im clippingfreien Modus 27,7%.

Maximum-Power-Saving-Modus

Der Fokus bei diesem Modus liegt, wie bereits gesagt, auf der Leistungersparnis. Clipping und eine geringe Verschlechterung der Bildqualität werden in Kauf genommen. Abbildung 8-13

zeigt einen Vergleich zwischen dem Originalbild und dem im Maximum-Power-Saving-Modus prozessierten Bild.

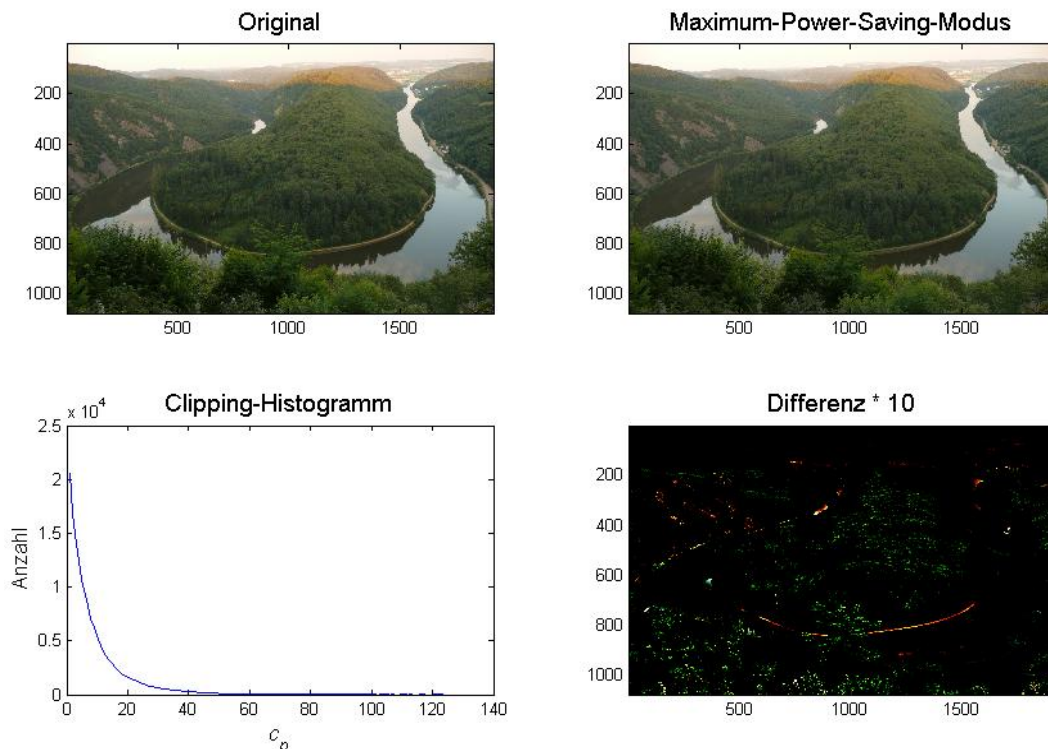


Abbildung 8-13: oben: Vergleich zwischen Originalbild (links) und Ergebnis der Prozessierung im Maximum-Power-Saving-Modus (rechts); unten: Clipping-Histogramm (links) und Differenz zwischen Original und prozessiertem Bild um Faktor 10 verstärkt (rechts)

Die visuelle Qualität des prozessierten Bildes hat sich etwas verschlechtert. Die Konturen der Baumkronen wirken bei genauem Hinsehen an manchen Stellen leicht unscharf. Der im Originalbild helle Weg am Rand des Flusses erscheint im prozessierten Bild etwas dunkler. Diese Beobachtungen spiegeln sich auch im Clipping-Histogramm und im Differenz-Bild wider. Deutlich mehr Pixel weisen Clipping auf. Um die Differenz besser sichtbar zu machen, wird diese um den Faktor zehn verstärkt dargestellt. Die im Vergleich zum clippingfreien Modus etwas schlechtere Bildqualität führt jedoch auch zu einer höheren Leistungersparnis (in diesem Beispiel 43,3%).

Automatischer Modus

Ziel der Prozessierung im automatischen Modus ist es, wie in Abschnitt 5.4.2 erläutert, die Leistungersparnis zu maximieren, ohne dass die visuelle Qualität darunter leidet. Dies wird am Beispiel in Abbildung 8-14 gezeigt.

Zur Festlegung des CAF werden vier Schwellwerte verwendet, wodurch fünf verschiedene Kondensierungsfunktionen zum Einsatz kommen (siehe Abbildung 8-14 oben links). Je heller ein hier dargestellter Pixel ist, desto weniger Leistung darf eingespart werden. Im Vergleich zum Ergebnis des Maximum-Power-Saving-Modus erkennt man hier eine deutlich bessere Bildqualität. Es sind mehr Details vorhanden, das Bild wirkt lebendiger. Dies spiegelt sich auch

im Clipping-Histogramm wider, aus dem man entnehmen kann, dass sich die Anzahl der clippingbehafteten Subpixel mehr als halbiert. Die Bildqualität ist tadellos und die Leistungersparnis beläuft sich auf 41,6%, womit diese nur knapp unter der des Maximum-Power-Saving-Modus liegt.

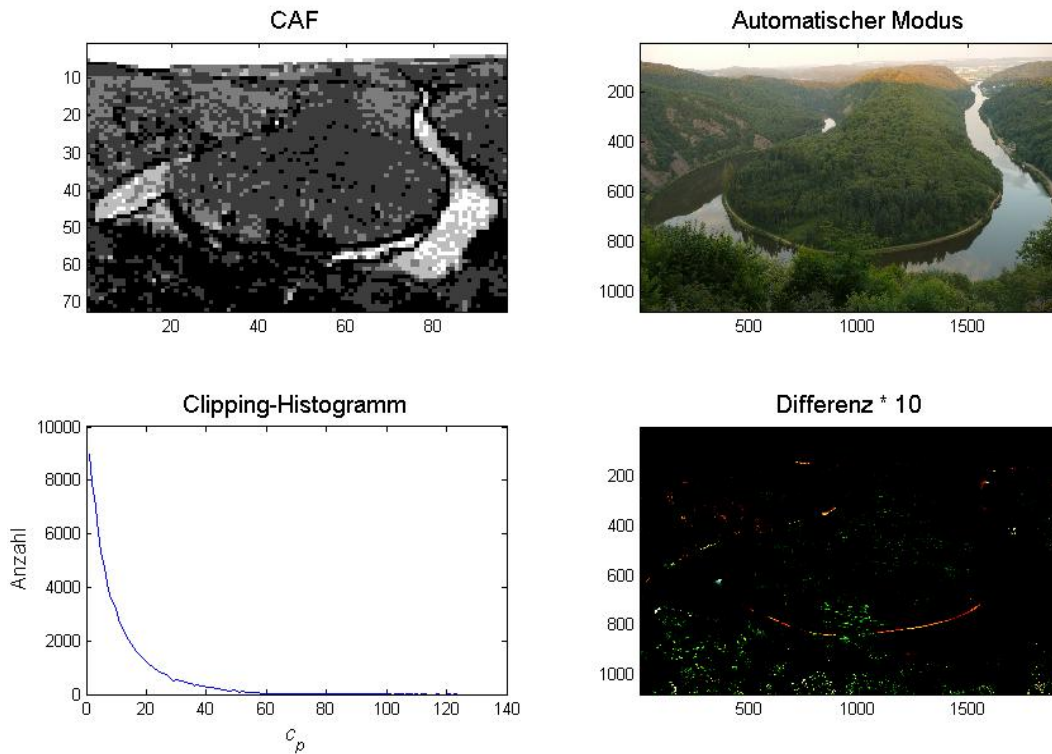


Abbildung 8-14: oben: CAF (links) und Ergebnis der Prozessierung im automatischen Modus (rechts); unten: Clipping-Histogramm (links) und Differenz zwischen Original und prozessiertem Bild um Faktor 10 verstärkt (rechts)

8.3.1.2 Edge-Lit TV

Clippingfreier Modus

Abbildung 8-15 zeigt den Vergleich zwischen Originalbild und dem Ergebnis der Prozessierung im clippingfreien Modus.

Die visuelle Qualität des prozessierten Bildes ist hervorragend, mit dem bloßen Auge lässt sich kein Unterschied zum Originalbild feststellen. Im Vergleich zum Ergebnis des clippingfreien Modus des Direct-Lit TVs (siehe Abbildung 8-12) fällt hier auf, dass insgesamt mehr Pixel nicht die vom Originalbild geforderte Helligkeit aufweisen, was zum einen auf die Mittelwertbildung bei der Kondensierung der LSFs und zum anderen auf die größere Zelle (60-60 Pixel) zurückzuführen ist. Die Leistungersparnis für das oben gezeigte Bild beträgt im clippingfreien Modus jedoch nur 9%, da die Granularität des Backlights mit seinen sechs LEDs sehr gering ist.

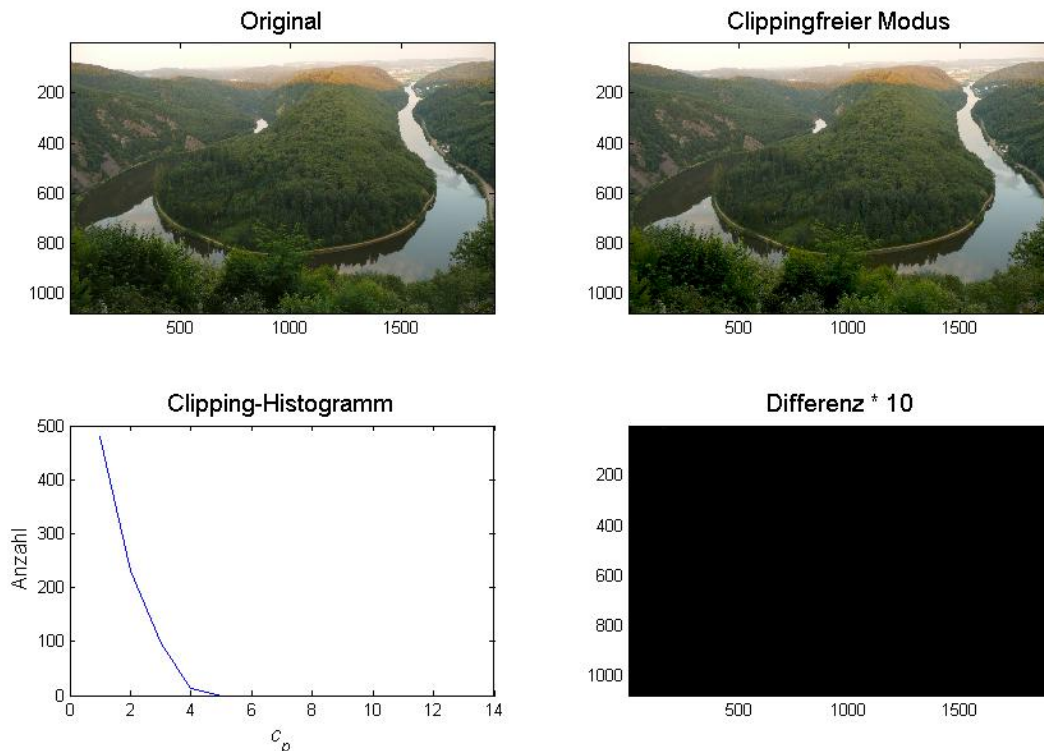


Abbildung 8-15: oben: Vergleich zwischen Originalbild (links) und Ergebnis der Prozessierung im clippingfreien Modus (rechts); unten: Clipping-Histogramm (links) und Differenz zwischen Original und prozessiertem Bild um Faktor 10 verstärkt (rechts)

Maximum-Power-Saving-Modus

Das Ergebnis des Maximum-Power-Saving-Modus wird in Abbildung 8-16 mit dem Originalbild verglichen.

Die Leistungersparnis für das im Maximum-Power-Saving-Modus prozessierte Bild beträgt 28,6%. Es fällt auf, dass sehr viele Pixel Clipping aufweisen, welches vornehmlich am oberen Rand des Bildes auftritt. Im prozessierten Bild ist diese clippingbehaftete Fläche zwar bei genauem Hinsehen und Vergleichen mit dem Originalbild erkennbar, ohne die Kenntnis des Originalbildes wäre das Clipping jedoch dem menschlichen Auge nicht aufgefallen. Somit ist die visuelle Qualität als sehr gut zu bewerten.

Generell lässt sich sagen, dass sich das Edge-Lit Display durch eine große Robustheit auszeichnet, da das Licht einer LED globaler verteilt wird und einen geringeren Gradienten hat. Dies führt zu einer Abschwächung der Wahrnehmung von Clipping.

Automatischer Modus

Das Ergebnis der Prozessierung im automatischen Modus ist in Abbildung 8-17 zu sehen.

Analog zum Beispiel in Abschnitt 8.3.1.1 werden zur Festlegung des CAF ebenfalls vier Schwellwerte verwendet. Je größer der CAF-Wert (je heller der dargestellte Pixel), desto weniger Leistung darf eingespart werden. Die visuelle Qualität des im automatischen Modus prozessierten Bildes ist hervorragend. Es tritt Clipping auf, welches aber im prozessierten Bild nicht negativ auffällt. Die Leistungersparnis in diesem Beispiel beträgt 21,8%.

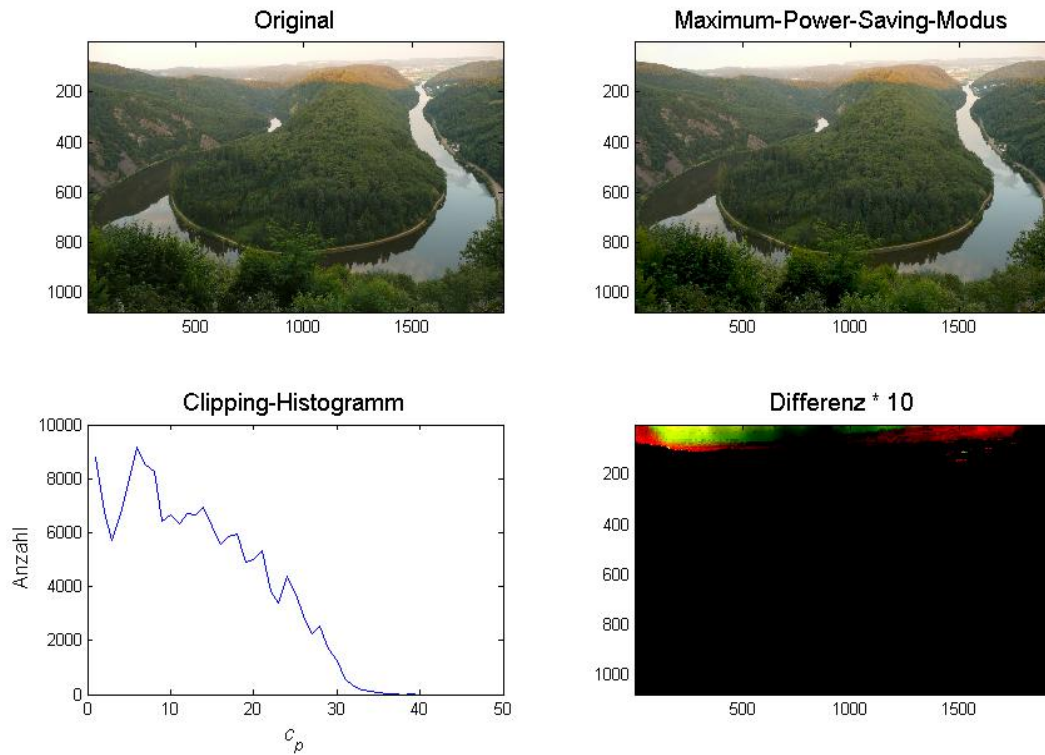


Abbildung 8-16: oben: Vergleich zwischen Originalbild (links) und Ergebnis der Prozessierung im Maximum-Power-Saving-Modus (rechts); unten: Clipping-Histogramm (links) und Differenz zwischen Original und prozessiertem Bild um Faktor 10 verstärkt (rechts)

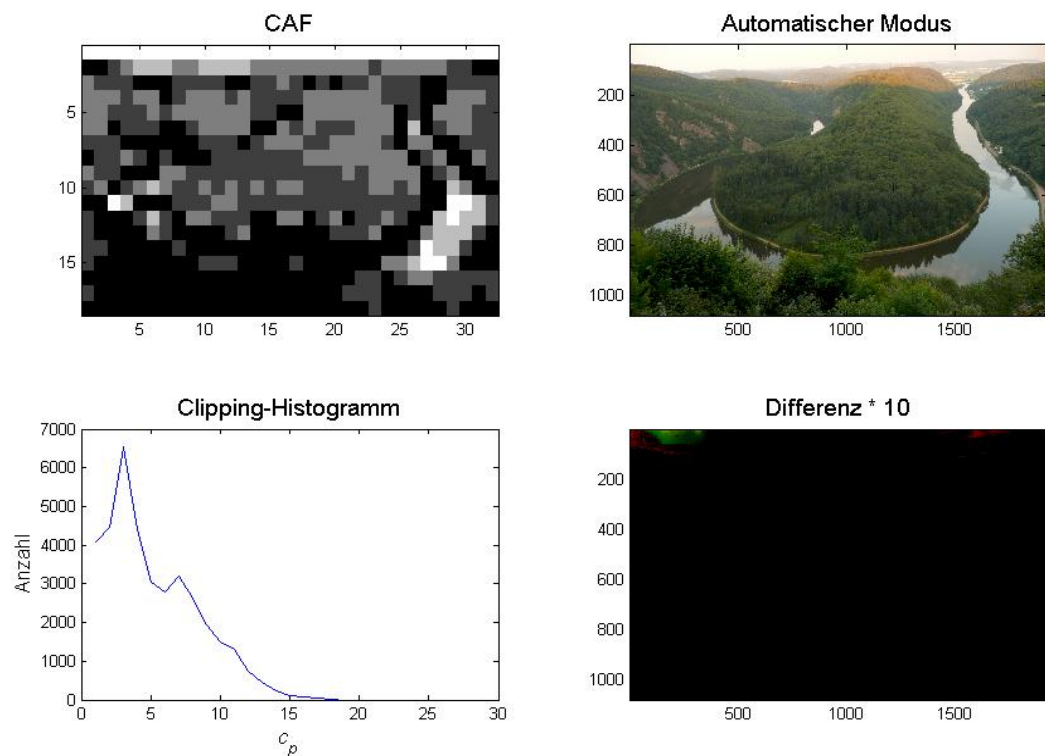


Abbildung 8-17: oben: CAF (links) und Ergebnis der Prozessierung im automatischen Modus (rechts); unten: Clipping-Histogramm (links) und Differenz zwischen Original und prozessiertem Bild um Faktor 10 verstärkt (rechts)

8.3.1.3 Direct-Lit-Signalanzeige

Clippingfreier Modus

Bedingt durch die Mittelwertbildung zur Ermittlung der kondensierten LSFs, kann auch bei der Signalanzeige vereinzelt leichtes Clipping auftreten. Da es sich hier jedoch um ein monochromes Display handelt, ist das Clipping aufgrund des großen Helligkeitsunterschiedes zwischen ein- und ausgeschalteter LC-Zelle nicht wahrnehmbar. Dies wird beim Betrachten von Abbildung 8-18 deutlich.

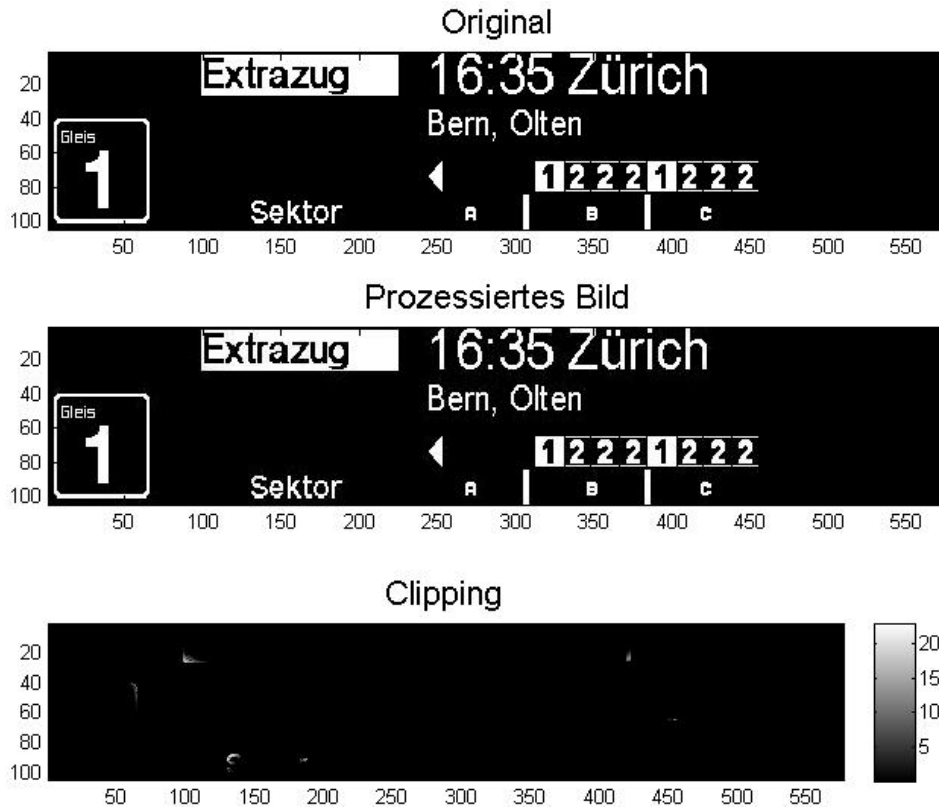


Abbildung 8-18: oben: Originalbild; Mitte: prozessiertes Bild; unten: Clipping

Der clippingfreie Modus liefert Ergebnisse von hervorragender visueller Qualität. Clipping tritt nur vereinzelt auf, und wenn, dann an Übergängen zwischen Schrift und Hintergrund, wo es aufgrund des großen Kontrastes nicht wahrnehmbar ist.

Variation von Δ_b

Im Folgenden werden die grafischen Ergebnisse der in Abschnitt 7.4.3.2 erwähnten Modifikation bei der Kondensierung untersucht. Zunächst wird anhand von Abbildung 8-19 gezeigt, dass Clipping mit steigendem Δ_b zunimmt.

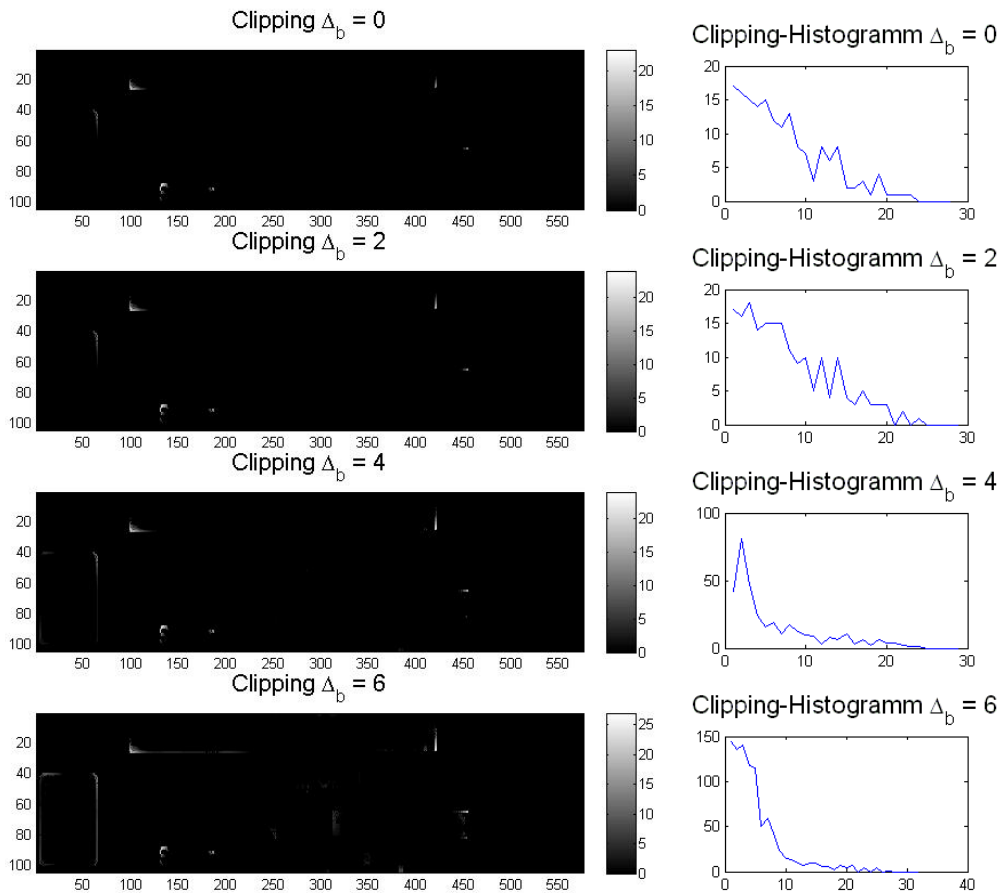


Abbildung 8-19: Vergleich des Auftretens von Clipping bei $\Delta_b = 0$, $\Delta_b = 2$, $\Delta_b = 4$ und $\Delta_b = 6$ – links: Darstellung des Clippings; rechts: Clipping-Histogramme

Man erkennt, dass bei $\Delta_b = 6$ am meisten Clipping auftritt, jedoch auch hier fast ausschließlich an den Übergängen zwischen Schrift und Hintergrund. Um eine Aussage über die visuelle Qualität des mit $\Delta_b = 6$ prozessierten Bildes zu treffen, wird dieses in Abbildung 8-20 mit dem Originalbild verglichen.

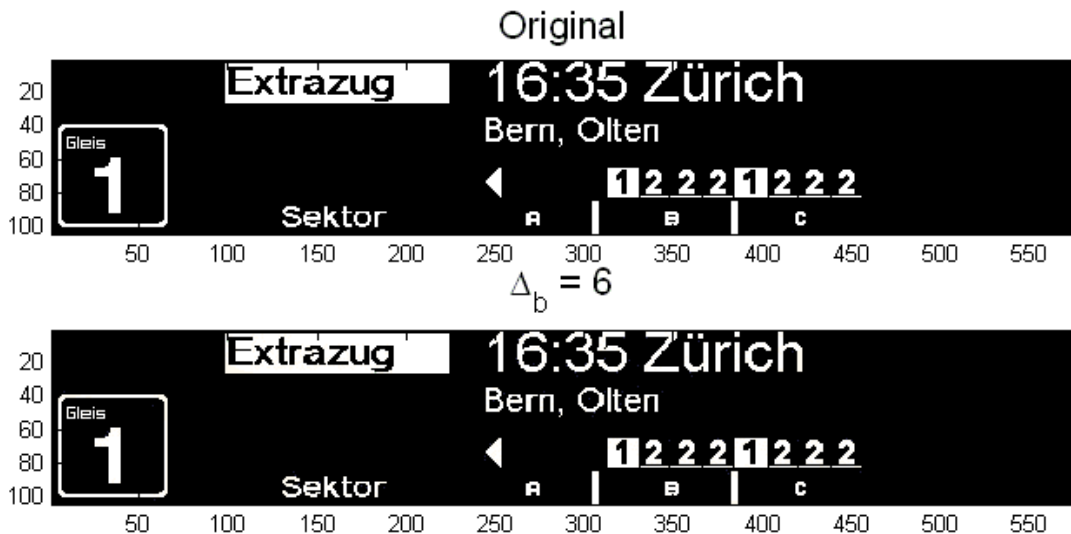


Abbildung 8-20: Vergleich des mit $\Delta_b = 6$ prozessierten Bildes mit dem Originalbild

Trotz des auftretenden Clippings kann das menschliche Auge keinen Unterschied zwischen den beiden Bildern ausmachen. Die visuelle Qualität leidet somit nicht unter dem erhöhten Clipping. Daher kann die präsentierte Methode verwendet werden, um noch mehr Leistung einzusparen.

8.3.2 Video

In Abschnitt 7.3.3 wird gezeigt, dass die LED-Werte bei Videoanwendungen immer erst um ein Frame verzögert vorliegen. Entweder verwendet man hier einen zusätzlichen Speicher in der Größe eines kompletten Bildes oder man passt die TFT-Werte des aktuellen Frames an die von den LED-Werten des vorangegangenen Frames erzeugte Helligkeit an.

Im ersten Fall entspricht die visuelle Qualität der unbewegter Bilder. Im zweiten Fall kann eine Verschlechterung der visuellen Qualität eintreten. Dies wird im Folgenden am Direct-Lit TV und Edge-Lit TV überprüft. Da beim Maximum-Power-Saving-Modus eine schlechtere visuelle Qualität zu erwarten ist als beispielsweise beim clippingfreien Modus, wird jeder der Condenser-Modi bezüglich der erreichten visuellen Qualität untersucht.

Die Signalanzeige wird nicht untersucht, da bei dieser keine Videofunktion gefordert wird.

Abbildung 8-21 zeigt die einzelnen Frames der untersuchten Videosequenz, welche Teil des IEC Testvideos ist.

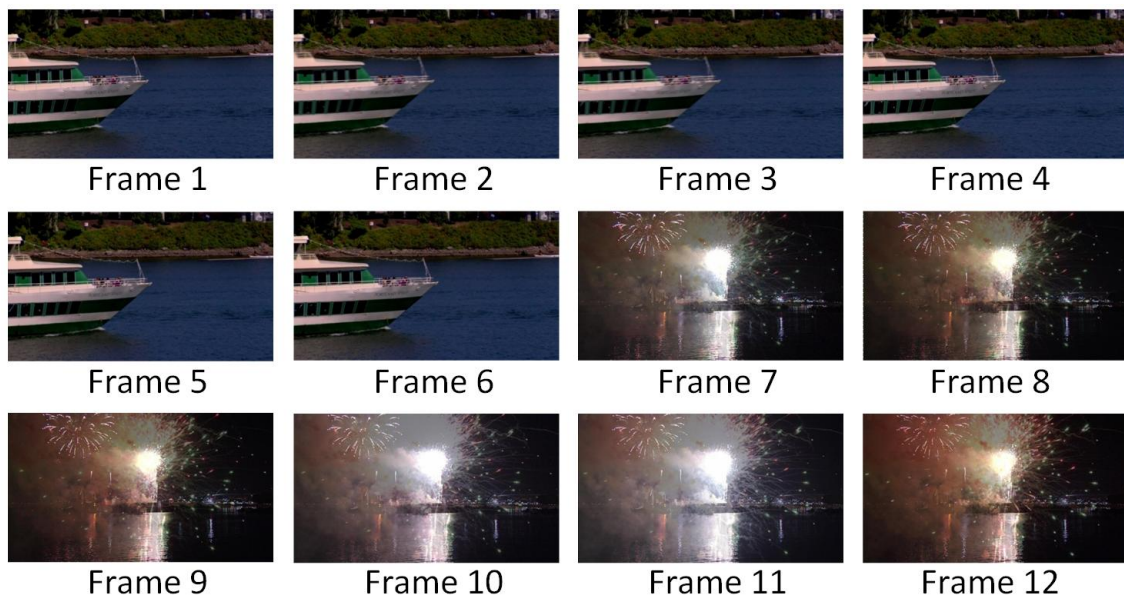


Abbildung 8-21: Frames der untersuchten Videosequenz

Besonderes Interesse weckt der Bildschnitt zwischen Frame 6 und Frame 7. Da sich die Bildinhalte von Frame 6 und Frame 7 sehr unterscheiden, ist hier die größte Abweichung zwischen Originalbild und prozessiertem Bild zu erwarten. Die LED-Werte des eher dunklen Frames 6 reichen nicht aus, um die vielen hellen Bereiche von Frame 7 ohne Clipping darzustellen.

Des Weiteren ist die Darstellung der Yacht in den Frames 1 bis 6 interessant, da diese der Berechnung der LED-Werte immer um einige Pixel vorseilt. Exemplarisch für diese sechs Frames wird der Übergang von Frame 3 zu Frame 4 näher untersucht.

Darüber hinaus können durch die schnellen Veränderungen des Bildinhaltes beim Feuerwerk in den Frames 7 bis 12 viele Artefakte und Detailverluste auftreten. Dies wird anhand des Überganges von Frame 9 zu Frame 10 analysiert.

Da für Frame 1 keine LED-Werte vorliegen, werden im Folgenden nur die Frames 2 bis 12 betrachtet.

8.3.2.1 Direct-Lit TV

Abbildung 8-22 vergleicht die visuelle Qualität von Frame 7 bei Variation der verschiedenen Condenser-Modi.

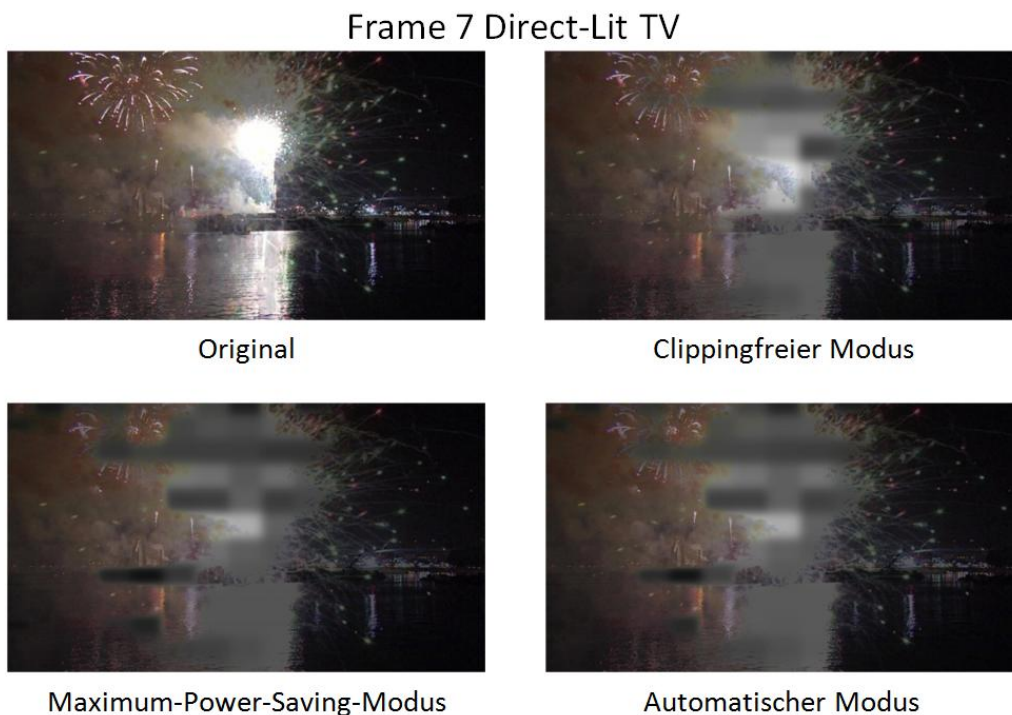


Abbildung 8-22: Vergleich der visuellen Qualität von Frame 7 bei Variation der Condenser-Modi

Selbst im clippingfreien Modus tritt bei diesem Bildschnitt großflächig Clipping auf. Abhilfe kann hier nur das Zwischenspeichern der Bilddaten über zwei Frames schaffen. Da die Zeitdauer für ein Frame bei Videoanwendungen zwischen 1/25 und 1/200 Sekunden liegt und der Bildinhalt sich hier komplett ändert, ist jedoch zu erwarten, dass das träge menschliche Auge diese Artefakte nicht erfassen kann.

Nun werden die Ergebnisse der Prozessierung von Frame 4 miteinander verglichen (siehe Abbildung 8-23).

Frame 4 Direct-Lit TV



Original



Clippingfreier Modus



Maximum-Power-Saving-Modus



Automatischer Modus

Abbildung 8-23: Vergleich der visuellen Qualität von Frame 4 bei Variation der Condenser-Modi

Bedingt durch die langsame Bewegung der Yacht, ist die visuelle Qualität der prozessierten Bilder mit der von unbewegten Bildern vergleichbar.

Anders verhält es sich in Frame 10 (siehe Abbildung 8-24).

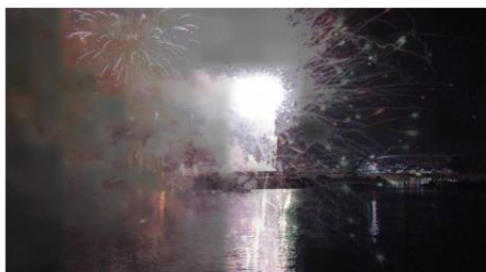
Frame 10 Direct-Lit TV



Original



Clippingfreier Modus



Maximum-Power-Saving-Modus



Automatischer Modus

Abbildung 8-24: Vergleich der visuellen Qualität von Frame 10 bei Variation der Condenser-Modi

Die visuelle Qualität im clippingfreien Modus ist sehr gut. Es tritt kein sichtbares Clipping auf. Im Maximum-Power-Saving-Modus ist jedoch eine deutliche Verschlechterung der visuellen Qualität zu erkennen. Hier sind die Gebiete um einige LEDs, die zur Ausleuchtung des vorherigen Frames nicht benötigt wurden, zu dunkel dargestellt.

Im automatischen Modus tritt dieses großflächige Clipping nicht auf. Die visuelle Qualität des prozessierten Bildes ist gut.

Abschließend wird das auftretende Clipping in Frame 2 bis 12 untersucht und mit dem Clipping verglichen, welches bei der Prozessierung der unbewegten Bilder auftritt. Für jeden Subpixel wird das Clipping c_p (siehe Formel 6.5) bestimmt, aufsummiert und durch die Anzahl aller Subpixel geteilt. Dadurch erhält man das Clipping/Pixel (siehe Formel 8.3):

$$\text{Clipping/Pixel} = \frac{\sum_{p=1}^{3 \cdot P} c_p}{3 \cdot P} \quad 8.3$$

Abbildung 8-25 vergleicht das Clipping/Pixel in den Frames 2 bis 12 bei Variation der verschiedenen Condenser-Modi.

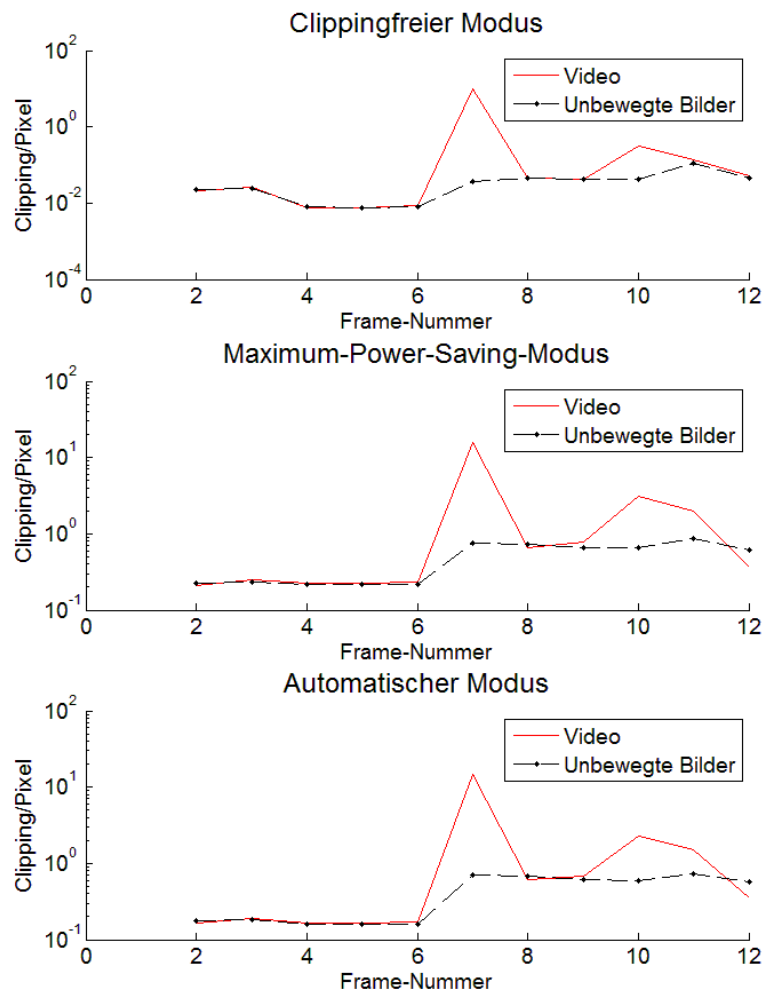


Abbildung 8-25: Clipping/Pixel in den verschiedenen Frames - von oben nach unten: clippingfreier Modus, Maximum-Power-Saving-Modus, automatischer Modus

Man erkennt, dass vor allem beim Bildschnitt zwischen Frame 6 und 7 sehr viel mehr Clipping auftritt, welches sich, wie in Abbildung 8-22 gezeigt wird, negativ auf die visuelle Qualität des dargestellten Bildes auswirkt. Da es sich jedoch um einen Bildschnitt handelt, kann angenommen werden, dass das Auge der schnellen Änderung nicht folgen kann. Das zwischen den Frames 9 bis 11 auftretende Clipping dagegen hat keine Auswirkung auf die visuelle Qualität (vergleiche Abbildung 8-24).

Es tritt also vermehrt dort Clipping auf, wo ein helles Frame auf ein dunkles Frame folgt. Ansonsten ist der Unterschied zwischen den Ergebnissen für unbewegte Bilder und für Video vernachlässigbar gering. Die in Abschnitt 7.3.3 präsentierte Vorgehensweise zur Behandlung von Video kann somit ohne Weiteres für das Direct-Lit TV angewendet werden.

8.3.2.2 Edge-Lit TV

Analog zu den visuellen Ergebnissen aus Abschnitt 8.3.2.1 werden nun die visuellen Ergebnisse des Edge-Lit TVs für Video untersucht. Abbildung 8-26 zeigt den Vergleich der visuellen Qualität von Frame 7 mit der des Originalbildes bei Variation der verschiedenen Condenser-Modi.

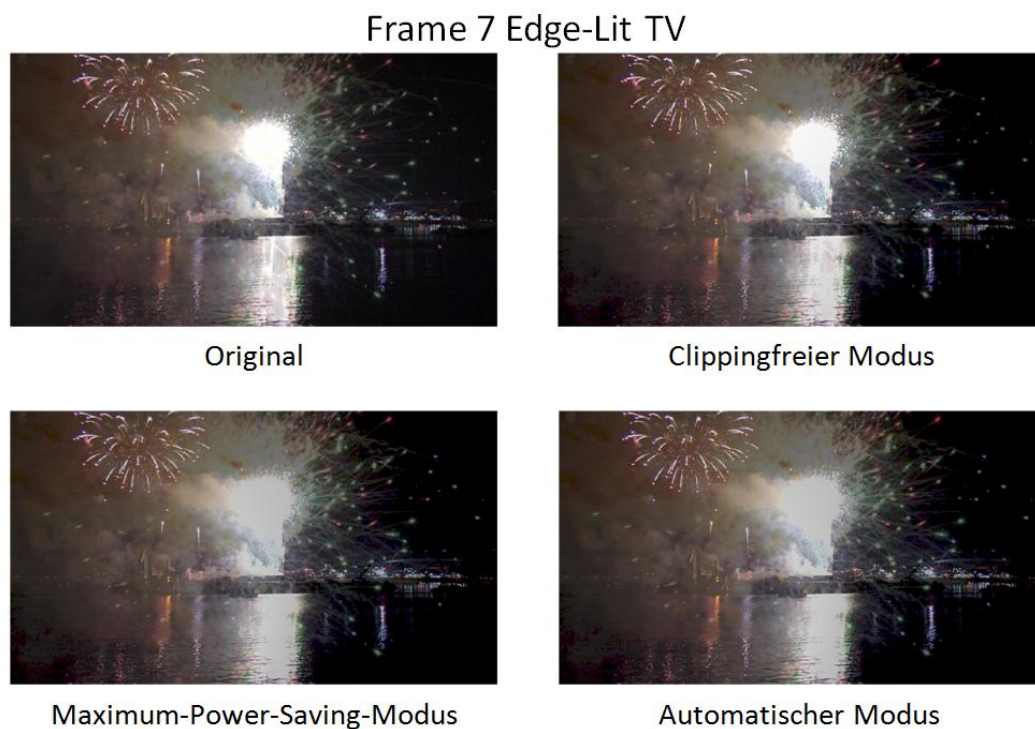


Abbildung 8-26: Vergleich der visuellen Qualität von Frame 7 bei Variation der Condenser-Modi

Im Vergleich zu den Ergebnissen aus Abbildung 8-22 haben die mit dem Edge-Lit TV prozessierten Bilder eine erheblich bessere Qualität. Einzig in der Mitte des Bildes erkennt man im Maximum-Power-Saving-Modus und im automatischen Modus großflächiges Clipping, welches jedoch wegen der kurzen Darstellungszeit für das menschliche Auge nicht bemerkbar ist.

Abbildung 8-27 vergleicht bei Variation der verschiedenen Condenser-Modi die visuelle Qualität von Frame 4 mit der des Originalbildes.

Frame 4 Edge-Lit TV



Original



Clippingfreier Modus



Maximum-Power-Saving-Modus



Automatischer Modus

Abbildung 8-27: Vergleich der visuellen Qualität von Frame 4 bei Variation der Condenser-Modi

Frame 10 Edge-Lit TV



Original



Clippingfreier Modus



Maximum-Power-Saving-Modus



Automatischer Modus

Abbildung 8-28: Vergleich der visuellen Qualität von Frame 10 bei Variation der Condenser-Modi

Alle drei dargestellten Modi liefern eine perfekte visuelle Qualität. Die kleinen Änderungen, verursacht durch die Vorwärtsbewegung der Yacht zwischen den Frames 1 bis 6, machen sich beim Edge-Lit TV im prozessierten Bild nicht bemerkbar.

Der Vergleich der visuellen Qualität von Frame 10 bei Variation der verschiedenen Condenser-Modi ist in Abbildung 8-28 zu sehen.

Genau wie im Beispiel aus Abbildung 8-27 ist die visuelle Qualität der mit allen drei Modi prozessierten Bilder, dank der bereits oben erwähnten Robustheit des Edge-Lit TVs, tadellos.

Abschließend wird (analog zu Abschnitt 8.3.2.1) das Clipping/Pixel für alle Frames ausgewertet (siehe Abbildung 8-29). Die Vorschrift zur Berechnung des Clippings/Pixel ist Formel 8.3 zu entnehmen.

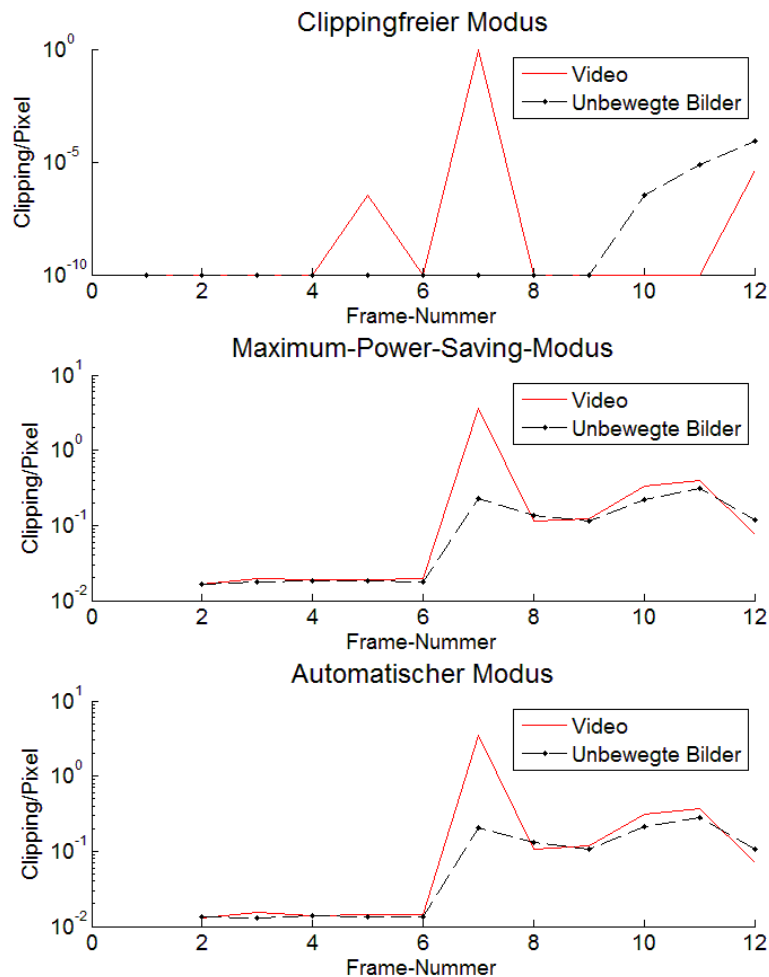


Abbildung 8-29: Clipping/Pixel in den verschiedenen Frames - von oben nach unten: clippingfreier Modus, Maximum-Power-Saving-Modus, automatischer Modus

Beim Edge-Lit TV tritt ebenfalls beim Bildschnitt zwischen Frame 6 und 7 vermehrt Clipping auf, welches sich jedoch, wie weiter oben gezeigt wurde, nicht auf die visuelle Qualität der

prozessierten Bilder auswirkt. Das in den anderen Frames auftretende Clipping ist vernachlässigbar gering.

Aufgrund der in diesem Abschnitt gezeigten Ergebnisse kann die in Abschnitt 7.3.3 präsentierte Vorgehensweise zur Behandlung von Video ohne Bedenken auch beim Edge-Lit TV angewendet werden.

8.3.3 Fazit

Im vorliegenden Abschnitt 8.3 wurde die visuelle Qualität der mit den verschiedenen Condenser-Modi prozessierten Bilder untersucht. Dabei wurden die Ergebnisse der Prozessierung sowohl von unbewegten als auch von bewegten Bildern miteinander verglichen. Die präsentierten Ergebnisse sind der Beweis dafür, dass die entwickelte Methode zur Berechnung von Local Dimming Backlight für reale Anwendungen geeignet ist.

Die visuelle Qualität der unbewegten Bilder ist hervorragend. Im clippingfreien Modus entspricht die visuelle Qualität des prozessierten Bildes exakt der des Originalbildes. Von Fall zu Fall kann leichtes Clipping auftreten, welches allerdings vom menschlichen Auge nicht wahrgenommen wird.

Im Maximum-Power-Saving-Modus wird ein erhöhtes Clipping zu Gunsten der höheren Leistungersparnis in Kauf genommen. Vereinzelt wirkt das prozessierte Bild flach und leblos, manche Details sind nicht so differenziert dargestellt wie im Originalbild. Das entstehende Clipping ist wahrnehmbar, jedoch wird es vom Betrachter nicht zwingend als störend empfunden. Ein drastischer Verlust an Bildinformation (siehe abhanden gekommene Objekte in Abbildung 2-15) ist auszuschließen.

Im automatischen Modus wird, wie in Abschnitt 5.4.2 bereits angekündigt, die Leistungersparnis maximiert, ohne dass dies Auswirkungen auf die visuelle Qualität des prozessierten Bildes hat. Die visuelle Qualität ist sehr gut, obwohl die Leistungersparnis im Mittel nur wenig unter der des Maximum-Power-Saving-Modus liegt.

Bei bewegten Bildern entspricht die visuelle Qualität in den meisten Fällen der unbewegter Bilder. Es spielt dabei keine Rolle, ob es sich um ein Direct-Lit TV oder ein Edge-Lit TV handelt oder welcher Condenser-Modus verwendet wird.

Lediglich bei einem Übergang von einem dunklen zu einem hellen Frame können beim Direct-Lit TV selbst im clippingfreien Modus Artefakte auftreten. Allerdings werden diese nur für ein Frame, also für eine sehr kurze Zeit, sichtbar. Möchte man diese Artefakte vollständig umgehen, so bleibt nur die Möglichkeit, die Bilddaten für ein Frame zwischenzuspeichern.

Das Edge-Lit TV ist auch bei Videoanwendungen robust gegen Clipping. Selbst bei einem Bildschnitt im Maximum-Power-Saving-Modus ist die visuelle Qualität gut. Der Grund hierfür liegt in der Beschaffenheit der LSFs. Eine helle LED sorgt für einen Helligkeitszuwachs in nahezu jedem Pixel. Ein heller Pixel sorgt bereits für mindestens eine helle LED. Kleine, stark

geclippte und somit für das Auge auffällige Bereiche sind daher beim Edge-Lit TV fast immer auszuschließen.

9 Zusammenfassung

In dieser Arbeit wird ein eigenständiger Weg zur Berechnung von Local Dimming LED Backlight präsentiert. Bei dem entwickelten Sorted-Sector-Covering-Algorithmus handelt es sich um einen Approximationsalgorithmus, der in der Lage ist, die LED-Werte für jede erdenkliche Backlight-Konstellation effizient in Echtzeit zu berechnen. Diese neuartige Methode basiert auf einem entwickelten mathematischen Modell von Local Dimming LED Backlight und zeichnet sich dadurch aus, dass die erzielten Ergebnisse frei von Clipping und sehr nahe an der theoretisch optimalen Lösung sind, womit eine fast maximale Leistungersparnis erreicht wird.

Mit dem eigens für SSC entwickelten Präprozessor, dem Condenser, der ein kontrolliertes Zulassen von Clipping ermöglicht, kann die Leistungersparnis noch weiter erhöht werden. Dabei können mit Hilfe von verschiedenen Condenser-Modi nicht nur Ergebnisse erzielt werden, die eine perfekte visuelle Qualität garantieren oder eine maximale Leistungersparnis erlauben, sondern auch einen auf die Bedürfnisse des Anwenders abgestimmten Kompromiss ermöglichen.

Über eine Analyse des Inhaltes der zu kondensierenden Zellen eines Bildes mit dem automatischen Condenser-Modus ist es möglich, die Leistungersparnis so weit zu maximieren, dass keine Verschlechterung der visuellen Qualität erkennbar ist.

Darüber hinaus wird die Anzahl der zu prozessierenden Pixel und somit die Laufzeit von SSC dank des Condensers auf einen sehr geringen Bruchteil reduziert. Bei alledem ist der zusätzliche Hardwareaufwand, der durch den Condenser entsteht, vernachlässigbar klein. Zudem fällt die Komplexität des SSC-Prozessors aufgrund der viel kleineren Anzahl der zu prozessierenden Pixel erheblich geringer aus.

Der entwickelte Local-Dimming-Prozessor kann effizient in Hardware, sowohl hardwired für ein ASSP/ASIC als auch in kostengünstige FPGAs, implementiert werden.

Es wurde an drei Prototypen gezeigt, dass der zusätzliche Speicherbedarf gering ist und sich die entwickelte Methode nahtlos in bestehende Displaycontroller integrieren lässt.

Somit konnte ein Verfahren zur Berechnung von Local Dimming LED Backlight entwickelt werden, das im Vergleich zu herkömmlichen, auf der Bildverarbeitung basierenden Methoden viele Vorteile bietet:

Durch den physikalischen Ansatz wird der Crosstalk zwischen den benachbarten LEDs bereits bei der Berechnung der LED-Werte berücksichtigt.

Der Optimierungsansatz garantiert, dass unerwünschtes Clipping vermieden wird und die Leistungersparnis nahe am Optimum ist.

Durch das kontrollierte Zulassen von Clipping mit Hilfe des Condensers gewährt die entwickelte Methode für die jeweilige Anwendung den besten Kompromiss zwischen Leistungseinsparung und Bildqualität. Sichtbare Artefakte und ein Verlust an Bildobjekten können vermieden werden.

Da SSC auf einem generischen Modell von Local Dimming Backlight basiert, werden keine Anforderungen bezüglich der Anordnung der LEDs gestellt. Daher kann die hier vorgestellte Methode in einem breiten Anwendungsfeld, wie beispielsweise in Edge-Lit LCDs, iPads, Notebooks, Smartphones etc., zur Energieeinsparung eingesetzt werden.

Darüber hinaus bildet SSC mit Bildkondensierung eine Grundlage für zukunftssträchtige Technologien wie 3D-Dimming und Field Sequential Color (siehe Abschnitte 10.1 und 10.2).

10 Ausblick

Diese Arbeit bildet die Grundlage für weitere Forschungen auf dem Gebiet von Local Dimming LED Backlight. Das vorliegende Kapitel versteht sich als Überblick zu sowohl bereits laufenden als auch zukünftigen Arbeiten am Lehrstuhl für Mikroelektronik der Universität des Saarlandes in diesem Bereich.

10.1 3D-Dimming

Unter dem Begriff 3D-Dimming versteht man lokales Dimmen (2D-Dimming) mit einem zusätzlichen Freiheitsgrad. Statt weißen LEDs verwendet man nun LEDs in den Farben Rot, Grün und Blau.

Beim 3D-Dimming muss man aufgrund der Beschaffenheit der Farbfilter und der abgestrahlten Wellenlänge des Lichtes das Übersprechen von LEDs einer Farbe in die Subpixel anderer Farben beachten. Dieses Phänomen wird als Farb-Crosstalk bezeichnet. Für echtes 3D-Dimming muss der Farb-Crosstalk nicht nur bei der Berechnung der TFT-Werte, sondern auch bei der Ermittlung der LED-Werte berücksichtigt werden, um ein Maximum an Leistung einzusparen. Gängige Ansätze, wie beispielsweise [Ch 07], sind dazu nicht in der Lage.

SSC dagegen berücksichtigt den Helligkeitsbeitrag benachbarter LEDs bei der Berechnung der LED-Werte, wodurch eine Adaption des Algorithmus an (echtes) 3D-Dimming möglich ist.

Um das mathematische Modell von 3D-Dimming Backlight zu erhalten, müssen Erweiterungen des Modells von Local Dimming Backlight (siehe Formel 3.3) vorgenommen werden, welche im Folgenden dargestellt werden.

Die erreichte Helligkeit b in einem Subpixel setzt sich nun aus den Werten der LEDs und deren Farb-Crosstalk in diesem Subpixel zusammen. Formel 10.1 zeigt dies am Beispiel eines roten Subpixels b_r , dessen Helligkeit nicht nur von der roten ($a_r^r \cdot x_r$), sondern auch von der grünen ($a_r^g \cdot x_g$) und der blauen ($a_r^b \cdot x_b$) LED bestimmt wird.

$$b_r = a_r^r \cdot x_r + a_r^g \cdot x_g + a_r^b \cdot x_b \quad 10.1$$

Das Gleichungssystem in Formel 10.2 beschreibt den Einfluss aller L roten LEDs auf alle $(3 \cdot P)$ Subpixel des LCDs.

$$\begin{bmatrix} a_{r1}^{r1} & a_{r1}^{r2} & \dots & a_{r1}^{rL} \\ a_{r2}^{r1} & & & \\ \dots & & & \\ a_{rp}^{r1} & & \dots & a_{rp}^{rL} \\ a_{g1}^{r1} & a_{g1}^{r2} & \dots & a_{g1}^{rL} \\ a_{g2}^{r1} & & & \\ \dots & & & \\ a_{gp}^{r1} & & \dots & a_{gp}^{rL} \\ a_{b1}^{r1} & a_{b1}^{r2} & \dots & a_{b1}^{rL} \\ a_{b2}^{r1} & & & \\ \dots & & & \\ a_{bp}^{r1} & & \dots & a_{bp}^{rL} \end{bmatrix} * \begin{bmatrix} x_{r1} \\ x_{r2} \\ \dots \\ x_{rL} \end{bmatrix} = \begin{bmatrix} b_{r1} \\ b_{r2} \\ \dots \\ b_{rp} \\ b_{g1} \\ b_{g2} \\ \dots \\ b_{gp} \\ b_{b1} \\ b_{b2} \\ \dots \\ b_{bp} \end{bmatrix} \quad \mathbf{10.2}$$

Hierbei bezeichnet a_{r2}^{r1} den Einfluss der ersten roten LED x_{r1} im zweiten roten Subpixel b_{r2} . Die erreichte Helligkeit im ersten roten Subpixel b_{r1} ergibt sich aus der Summe der Beiträge aller L roten LEDs. Die Anzahl der Einträge der Einflussmatrix hat sich um den Faktor drei vergrößert.

Erweitert man nun Formel 10.2 um die Information aus Formel 10.1, so erhält man das mathematische Modell für 3D-Dimming.

$$\begin{bmatrix} a_{r1}^{r1} & a_{r1}^{r2} & \dots & a_{r1}^{rL} & a_{r1}^{g1} & a_{r1}^{g2} & \dots & a_{r1}^{gL} & a_{r1}^{b1} & a_{r1}^{b2} & \dots & a_{r1}^{bL} \\ a_{r2}^{r1} & & & & a_{r2}^{g1} & & & & a_{r2}^{b1} & & & \\ \dots & & & & \dots & & & & \dots & & & \\ a_{rp}^{r1} & & & a_{rp}^{rL} & a_{rp}^{g1} & & & a_{rp}^{gL} & a_{rp}^{b1} & & & a_{rp}^{bL} \\ a_{g1}^{r1} & a_{g1}^{r2} & \dots & a_{g1}^{rL} & a_{g1}^{g1} & a_{g1}^{g2} & \dots & a_{g1}^{gL} & a_{g1}^{b1} & a_{g1}^{b2} & \dots & a_{g1}^{bL} \\ a_{g2}^{r1} & & & & a_{g2}^{g1} & & & & a_{g2}^{b1} & & & \\ \dots & & & & \dots & & & & \dots & & & \\ a_{gp}^{r1} & & & a_{gp}^{rL} & a_{gp}^{g1} & & & a_{gp}^{gL} & a_{gp}^{b1} & & & a_{gp}^{bL} \\ a_{b1}^{r1} & a_{b1}^{r2} & \dots & a_{b1}^{rL} & a_{b1}^{g1} & a_{b1}^{g2} & \dots & a_{b1}^{gL} & a_{b1}^{b1} & a_{b1}^{b2} & \dots & a_{b1}^{bL} \\ a_{b2}^{r1} & & & & a_{b2}^{g1} & & & & a_{b2}^{b1} & & & \\ \dots & & & & \dots & & & & \dots & & & \\ a_{bp}^{r1} & & & a_{bp}^{rL} & a_{bp}^{g1} & & & a_{bp}^{gL} & a_{bp}^{b1} & & & a_{bp}^{bL} \end{bmatrix} * \begin{bmatrix} x_{r1} \\ x_{r2} \\ \dots \\ x_{rL} \\ x_{g1} \\ x_{g2} \\ \dots \\ x_{gL} \\ x_{b1} \\ x_{b2} \\ \dots \\ x_{bL} \end{bmatrix} = \begin{bmatrix} b_{r1} \\ b_{r2} \\ \dots \\ b_{rp} \\ b_{g1} \\ b_{g2} \\ \dots \\ b_{gp} \\ b_{b1} \\ b_{b2} \\ \dots \\ b_{bp} \end{bmatrix} \cong \begin{bmatrix} i_{r1} \\ i_{r2} \\ \dots \\ i_{rp} \\ i_{g1} \\ i_{g2} \\ \dots \\ i_{gp} \\ i_{b1} \\ i_{b2} \\ \dots \\ i_{bp} \end{bmatrix} \quad \mathbf{10.3}$$

- a_{b2}^{r1} : Einfluss der ersten roten LED im zweiten blauen Subpixel
- x_{r2} : LED-Wert der zweiten roten LED
- b_{r1} : erreichte Helligkeit im ersten roten Subpixel
- i_{r1} : Helligkeit des roten Subpixels im ersten Pixel

Die Anzahl der Einträge der Einflussmatrix hat sich durch die Berücksichtigung des Farbcrosstalks und durch die Verwendung von roten, grünen und blauen LEDs um das Neunfache erhöht.

Durch den Farb-Crosstalk gestaltet sich auch die Adaption der TFT-Werte schwieriger als beim Local Dimming, da eine Veränderung des TFT-Wertes im roten Subpixel auch zu einer Veränderung des grünen und blauen Farbanteils des Pixels führt.

Aus diesem Grund müssen die TFT-Werte für alle drei Subpixel in Abhängigkeit zueinander berechnet werden. Das führt für jeden Pixel zu einem linearen Gleichungssystem mit den drei TFT-Werten t_r , t_g und t_b der Subpixel als Unbekannte (siehe Formel 10.4).

$$\begin{pmatrix} b_r^r & b_g^r & b_b^r \\ b_r^g & b_g^g & b_b^g \\ b_r^b & b_g^b & b_b^b \end{pmatrix} * \begin{pmatrix} t_r \\ t_g \\ t_b \end{pmatrix} = \begin{pmatrix} i_r \\ i_g \\ i_b \end{pmatrix} \quad 10.4$$

Hierbei ist b_b^g der Helligkeitsbeitrag aller grünen LEDs im betrachteten blauen Subpixel.

Die Anpassung von SSC an 3D-Dimming stellt eine Herausforderung dar.

Speicherbedarf und -zugriff sind aufgrund der Komplexität der Einflussmatrix erhöht. Um möglichst viel Leistung einsparen zu können, muss die Prozessierung dahingehend erweitert werden, dass bei der Festlegung der LED-Werte der Farb-Crosstalk der anderen LEDs berücksichtigt wird. Auch die Ermittlung der TFT-Werte gestaltet sich, wie in Formel 10.4 zu sehen ist, schwieriger als zuvor. Es sind weitere intensive Forschungsarbeiten erforderlich.

10.2 Field Sequential Color

Eine der vielversprechendsten und neuesten Entwicklungen im Bereich der LCDs ist Field Sequential Color (FSC) (siehe ([Che 05], [Hsi 08] und [Yam 02])). Hierbei werden die Farbfilter der LCDs (siehe Abschnitt 2.2) entfernt. Farben werden nur noch durch Verwendung von roten, grünen und blauen LEDs erzeugt. Durch das Weglassen der Farbfilter wird die Lichtausbeute des LCDs um zwei Drittel erhöht. Dadurch kann beim Dimmen des Backlights noch erheblich mehr Leistung eingespart werden als beim 2D- bzw. 3D-Dimming. Darüber hinaus sinken durch das Weglassen der Farbfilter auch die Herstellungskosten des Displays.

Bei FSC werden die farbigen LEDs, nicht wie bisher gleichzeitig, sondern nacheinander ein- und ausgeschaltet. Jede Farbe erhält ein eigenes Zeitfenster. Diese Zeitfenster werden von kurzen Schwarzphasen unterbrochen, in denen die TFT-Werte neu eingestellt werden. Abbildung 10-6 zeigt den zeitlichen Verlauf von FSC.

Eine Adaption des SSC-Algorithmus an diese neuartige Technologie stellt keine Probleme dar. Das Hauptproblem bei FSC liegt aktuell in der Geschwindigkeit der LC-Zelle, da diese, wie in Abbildung 10-1 zu erkennen ist, dreimal schneller sein muss als bei herkömmlichen Technologien.

Ein weiteres Problem von FSC ist der Color-Breakup [Lin 09]. Dieses Problem könnte durch eine Kombination von SSC und dem ebenfalls am Lehrstuhl für Mikroelektronik an der Universität des Saarlandes erfundenen CMLA [Xu 08] behoben werden.

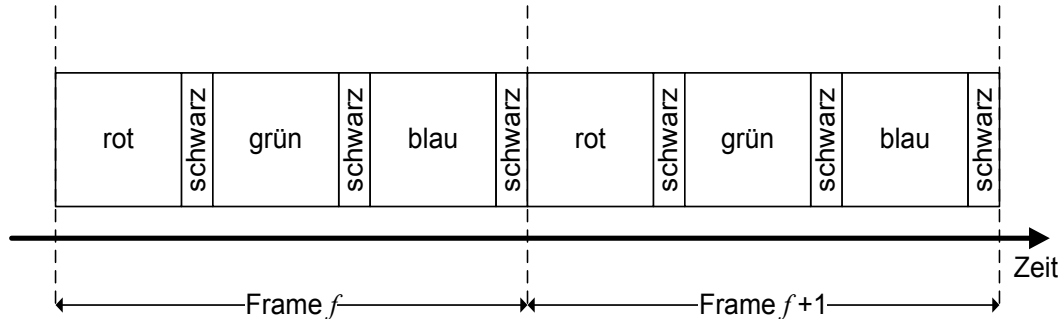


Abbildung 10-1: Zeitlicher Verlauf von FSC

10.3 Analyse des Zellhistogramms zur Verbesserung des automatischen Condenser-Modus

Der in Abschnitt 5.4.2 präsentierte automatische Condenser-Modus führt, wie in Kapitel 8.2 gezeigt, bereits zu einer hohen Leistungersparnis bei perfekter visueller Qualität. Im Folgenden wird ein erster Ansatz gezeigt, wie die Leistungersparnis bei gleichbleibender Bildqualität noch weiter erhöht werden kann. Dabei wird das Histogramm der zu kondensierenden Zelle mit in die Berechnung des kondensierten Pixels einbezogen.

In der Bildverarbeitung versteht man unter einem Histogramm die statistische Verteilung der Pixelwerte eines Bildes. Hierbei handelt es sich um einen Punktoperator, weshalb geläufige Histogramm-Operationen, wie beispielsweise die in Abbildung 2-13 gezeigte Histogramm-Streckung, jeweils über das komplette Bild (global) ausgeführt werden.

Es wurden erste Versuche gemacht, das Histogramm lokal auf Condenser-Zellebene zu analysieren und die Ergebnisse in die Kondensierung einfließen zu lassen.

Eine einfache Methode wird nun anhand eines Beispiels skizziert. Abbildung 10-2 zeigt einen Ausschnitt aus dem in Abbildung 5-13 (linke Seite) dargestellten Bild.



Abbildung 10-2: Bildausschnitt zur Analyse des Zellhistogramms

Zur besseren Unterscheidung der Zellen wurden diese in der Abbildung nachträglich durch schwarze Balken voneinander abgetrennt. Die Zellen werden von links oben nach rechts unten durchnummeriert. Abbildung 10-3 zeigt die Histogramme der vier Zellen.

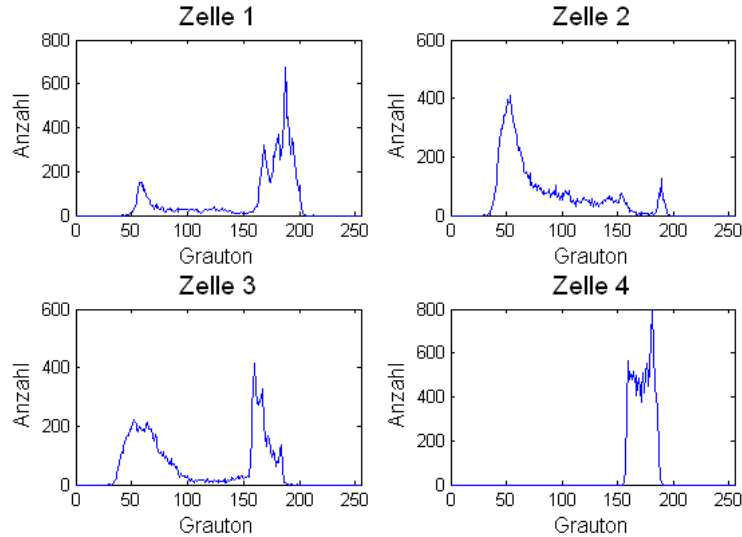


Abbildung 10-3: Histogramme der Zellen des Bildausschnittes

Jedes der lokalen Histogramme wird nun nach dem folgenden Schema bearbeitet:

Es werden die größten Pixelwerte so lange aus dem Histogramm entfernt, bis entweder die vorher festgelegte, maximale Anzahl von entfernten Pixeln oder die vorher festgelegte maximale Differenz zwischen hellstem und aktuellem Pixel erreicht ist.

Abbildung 10-4 zeigt die Histogramme aus Abbildung 10-3 in der beschriebenen Modifizierung. Man erkennt, dass in jedem Histogramm eine gewisse Anzahl der hellsten Pixel entfernt wurde.

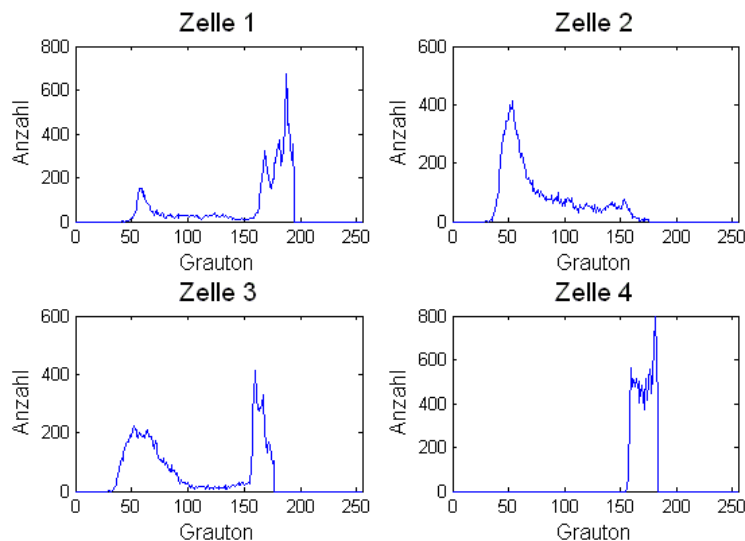


Abbildung 10-4: Histogramme der Zellen des Bildausschnittes nach der Modifizierung

Aus den beiden ermittelten Werten (MAX, AMW), die dem modifizierten Histogramm entnommen werden, wird dann der Wert des kondensierten Pixels bestimmt.

Allerdings ist bei der Verwendung dieser Methode Vorsicht geboten. Durch das Entfernen unterschiedlich vieler Werte bei benachbarten Zellen können, gerade bei Direct-Lit LCDs, Artefakte auftreten (siehe Abbildung 10-5).

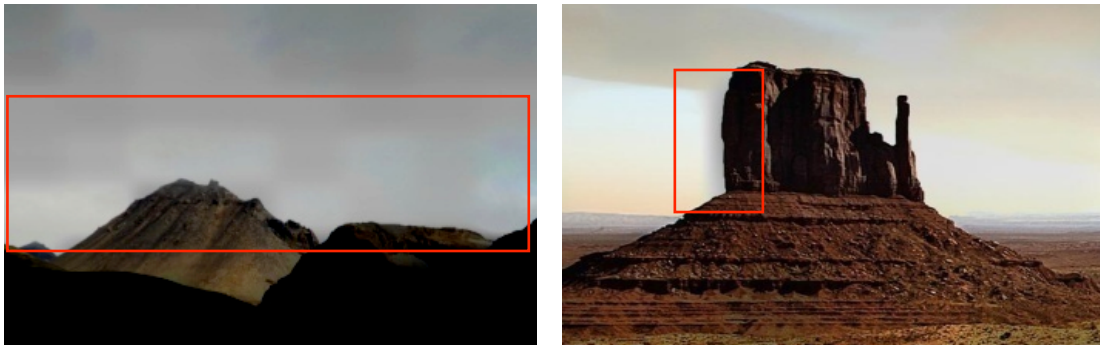


Abbildung 10-5: Zwei Beispiele für entstehende Artefakte bei der lokalen Histogrammanalyse

Erste Versuche haben zu einer Erhöhung der in den Abschnitten 8.2.1.1 und 8.2.1.2 gezeigten Leistungersparnis des automatischen Modus um durchschnittlich 3% geführt, ohne dass die Bildqualität dabei merklich verschlechtert wurde. Der zusätzlich benötigte Speicher zum Ablegen der Histogrammdaten ist gering.

10.4 Unterdrückung von LED-Flicker mit SSC

Wenn sich kleine Bereiche von Pixeln eines Frames schnell ändern, während andere Bereiche unverändert bleiben, so können sich ebenfalls die Werte der LEDs schnell ändern, die diese Bereiche dominieren. Abbildung 10-6 verdeutlicht dies am Beispiel eines sich bewegenden Objektes.

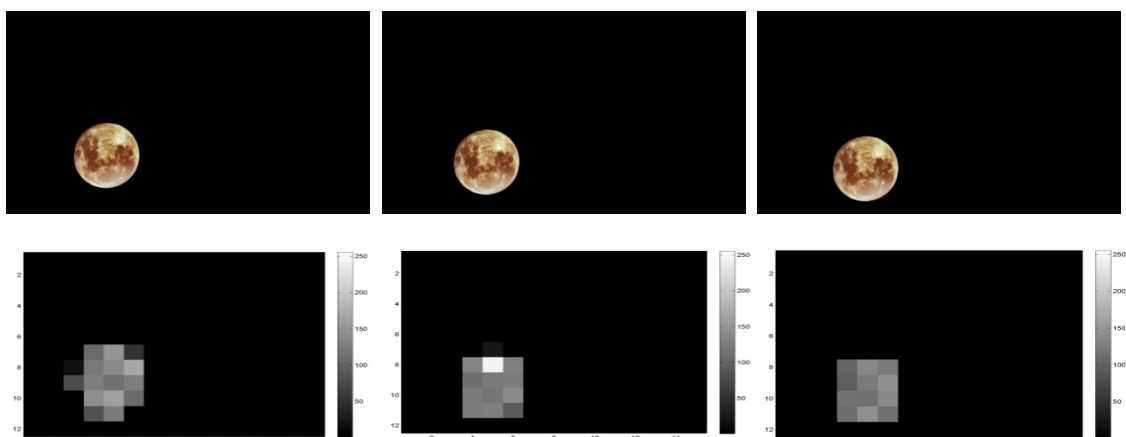


Abbildung 10-6: Sich bewegendes Objekt (oben) und die dazugehörigen LED-Werte (unten)

Die visuelle Qualität jedes einzelnen der prozessierten Frames des sich bewegenden Objektes ist tadellos. Betrachtet man jedoch die Einzelbilder in schneller Abfolge, kann es vorkommen, dass die Helligkeit einzelner LEDs zwischen mehreren Frames stark schwankt. Um dies zu verdeutlichen, muss man die im unteren Teil der Abbildung 10-6 gezeigten LED-Werte der einzelnen Frames betrachten. Es fällt auf, dass eine der LEDs im mittleren dargestellten Frame deutlich heller ist als in den beiden anderen Frames. Es tritt an dieser Stelle ein Phänomen auf, das LED-Flicker genannt wird. Dieses äußert sich in Helligkeitsschwankungen der LEDs, die vom menschlichen Auge als störend empfunden werden.

Anhand des in diesem Beispiel gezeigten LED-Flickers wird deutlich, dass es notwendig sein kann, den Verlauf der LED-Werte gewissen Einschränkungen zu unterwerfen. Ein zukünftiges Projekt ist es daher, die LED-Werte der vorangegangenen Frames bei der Berechnung von SSC zu berücksichtigen, um so LED-Flicker zu vermeiden.

In einem ersten Ansatz könnte das Problem so gelöst werden, dass das aktuelle Frame ausgehend von der Lösung des vorangegangenen Frames berechnet wird. Dadurch würden die LB von SSC entfallen. Gleichzeitig müsste die Möglichkeit gegeben sein, einzelne LED-Werte zu reduzieren. Dies könnte in Form eines Zwischenschrittes in der IP des Algorithmus stattfinden.

Eine weitere Möglichkeit zur Unterdrückung von LED-Flicker wäre, bei der Bestimmung der LED-Werte nicht nur auf eine möglichst hohe Leistungseinsparung zu achten, sondern auch den vorherigen LED-Wert zu berücksichtigen. Dabei erhalten sowohl die Summe der LED-Werte (Leistungersparnis) als auch der Unterschied der LED-Werte zum vorherigen Frame eine Gewichtung. Dadurch wird gewährleistet, dass sich LED-Werte aufeinanderfolgender Bilder nicht zu sehr voneinander unterscheiden.

An dieser Stelle muss jedoch angemerkt werden, dass LED-Flicker bei SSC nur bei Direct-Lit Backlights, und auch hier erheblich seltener als bei gängigen Methoden zur Berechnung von Local Dimming Backlight, wahrnehmbar ist.

10.5 Stabilitätsbetrachtung

Bei der Stabilitätsbetrachtung wird untersucht, welchen Schwankungen das Backlight unterliegen kann, ohne dass dies Einfluss auf die visuelle Qualität des prozessierten Bildes hat. Folgende vier Faktoren haben einen Einfluss auf die Stabilität des Backlights:

- **Diskrepanz bzgl. LED-Helligkeit** = Abweichung der LSF von einer LED zur anderen LED
- **Diskrepanz bzgl. LED-Farbe** = Abweichung der Farbtemperatur von einer LED zu einer anderen LED
- **Platzierungsgenauigkeit der LEDs** = Genauigkeit, mit der die LEDs auf der Platine positioniert sind

- **Diskrepanz bzgl. LED-Treiber** = unterschiedliche Ströme und somit unterschiedliche Helligkeiten der LEDs

Eine Gleichbetrachtung der LEDs hat mehrere Vorteile. So müssen die Helligkeitsverteilungen der LEDs nicht für jedes Display neu vermessen werden. Darüber hinaus bedeutet dies, dass weniger Speicher für die Einflussmatrix benötigt wird, wodurch die beiden in Abschnitt 7.2.1.4 beschriebenen LSF-RAMs noch weiter reduziert werden können und somit die Implementierung noch effizienter gestaltet werden kann.

Erste Versuche haben ergeben, dass eine relative Abweichung von 3% bei einem einzelnen der vier oben genannten Faktoren tolerierbar ist.

10.6 Metrik für Bildqualität

In einer Webstudie wird die visuelle Qualität verschiedener mit Condenser, SSC und Postprozessor prozessierter Bilder von Probanden bewertet. Damit ist es zum einen möglich, den Condenser noch weiter in Bezug auf Leistungersparnis und visuelle Qualität zu optimieren. Zum anderen kann mit den Resultaten der Webstudie eine Metrik für die visuelle Qualität eines prozessierten Bildes entwickelt werden.

In dieser Metrik gilt es, den Zusammenhang, in dem das Clipping auftritt, sprich die Beschaffenheit der umliegenden Pixelwerte, bei der Charakterisierung des Clippings mit einzubeziehen, da Clipping nur in Kombination mit den umliegenden Pixeln wahrgenommen wird.

Dem Autor dieser Arbeit ist eine einzige Veröffentlichung zu den Auswirkungen von Clipping auf die visuelle Qualität bekannt [siehe Ch 10]. Jedoch wird dort das Problem des Clippings nur auf den clippingbehafteten Pixel selbst bezogen.

Appendix A – Gaußscher Tiefpass im Frequenzbereich

Zunächst wird das Bild vom Zeitbereich mit der diskreten Fourier-Transformation in den Frequenzbereich übertragen:

$$F(u, v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) \cdot e^{-j2\pi \left(\frac{ux}{M} + \frac{vy}{N} \right)} \quad \text{A.1}$$

Ein Gaußscher Tiefpass im Frequenzbereich hat die folgende Form:

$$H(u, v) = 1 - e^{-\frac{D^2(u, v)}{2D_0^2}} \quad \text{A.2}$$

$D(u, v)$ beschreibt die Entfernung zum Ursprung $(u, v) = (M/2, N/2)$ des Bildes.

$$D(u, v) = \sqrt{\left(u - \frac{M}{2}\right)^2 + \left(v - \frac{N}{2}\right)^2} \quad \text{A.3}$$

Im Gegensatz zu einem idealen Tiefpassfilter, bei dem die Grenzfrequenz D_0 alle über ihr liegenden Frequenzanteile abschneidet, bestimmt D_0 beim Gaußschen Tiefpass die Steilheit der Flanken des Tiefpasses (siehe Abbildung A-1).

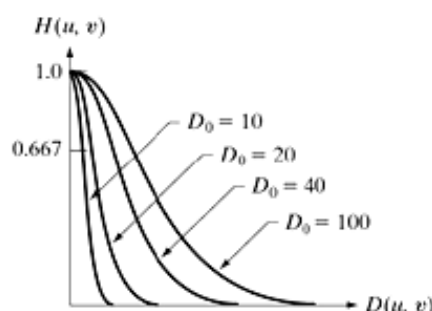


Abbildung A-1: Einfluss der Grenzfrequenz auf die Steilheit der Flanken des Tiefpasses [aus GW 08]

Nun wird das neue gefilterte Bild berechnet:

$$F_{TP}(u, v) = F(u, v) \cdot H(u, v) \quad \text{A.4}$$

Schließlich wird mit Hilfe der diskreten inversen Fouriertransformation das gefilterte Bild $F_{TP}(u, v)$ wieder in den Zeitbereich transformiert.

$$F(x, y) = \frac{1}{M \cdot N} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F_{TP}(u, v) \cdot e^{j2\pi \left(\frac{ux}{M} + \frac{vy}{N} \right)} \quad \text{A.5}$$

Für eine ausführliche Beschreibung der hier gezeigten Filterung siehe [GW 08].

Appendix B - Treiben einer LED mit PWM

Die LEDs eines Local Dimming Backlights werden für gewöhnlich mit Pulsweitenmodulation (PWM) betrieben. Dabei wird bei einer konstanten Frequenz das Verhältnis zwischen Zu- und Wegschalten eines konstanten Stromes eingestellt. Abbildung B-1 veranschaulicht dies grafisch.

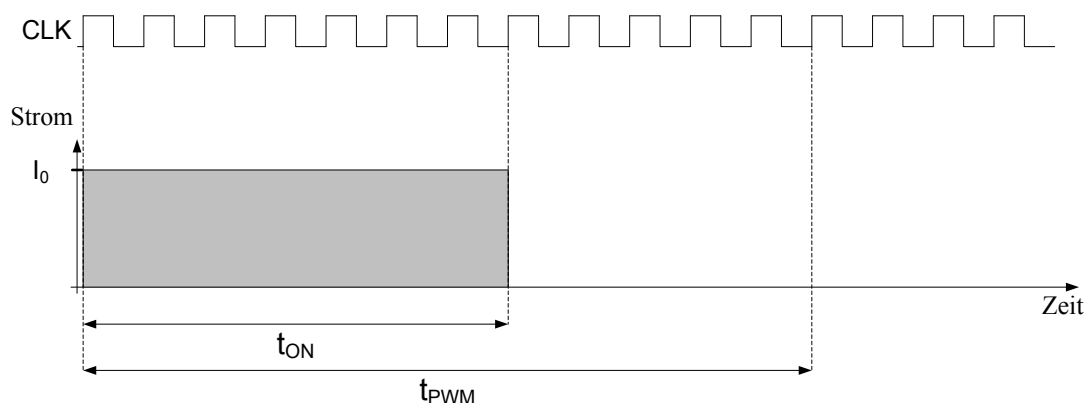


Abbildung B-1: PWM-Verhältnis

Ein PWM-Zyklus t_{PWM} dauert hier 12 CLK-Zyklen. Von diesen 12 CLK-Zyklen wird der Strom I_0 für eine festgelegte Zeit t_{ON} zugeschaltet. Im Kontext dieser Arbeit bedeutet dies, dass die mit diesem Signal betriebene LED $x(k)$ auf den Wert $7/12$ eingestellt wird.

Ein LED-Treiber hat den in Abbildung B-2 gezeigten Aufbau:

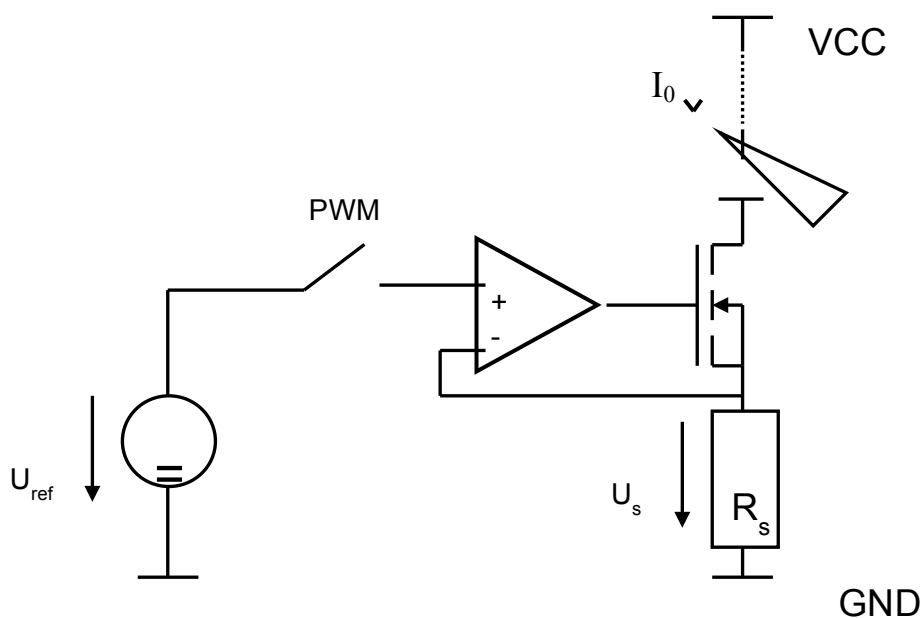


Abbildung B-2: Aufbau eines LED-Treibers

Die Spannung U_{ref} bestimmt, gemeinsam mit dem Tastverhältnis t_{ON}/t_{PWM} , die Helligkeit der LEDs. Der Operationsverstärker versucht seine beiden Eingänge anzugleichen, wodurch über

dem Shunt-Widerstand R_s die Spannung $U_s = U_{ref}$ abfällt. Die Größe des Widerstandes R_s lässt sich über U_s und den gewünschten Strom I_0 festlegen. Für eine ausführliche Beschreibung von PWM siehe [KS 00].

Appendix C - Bildabhängige Prozessierungsreihenfolge der Pixel in der IP

Im Folgenden wird eine weitere, zeit- und HW-intensivere Vorgehensweise zur Berechnung der IP präsentiert. Hierbei wird zunächst die durch die LB erreichte Helligkeit b_p in jedem Pixel berechnet und mit der geforderten Helligkeit i_p verglichen. Liegt ein Helligkeitsdefizit vor, merkt man sich den entsprechenden Pixel und dessen Defizit. Wenn alle Pixel betrachtet wurden, wird die dominierende LED des Pixels mit dem größten Defizit erhöht. Würde man die LED, wie bei der Berechnung der FDC, um die gesamte Differenz

$$\Delta x(k) = \frac{i_p - \sum_{\ell=1}^L a_p(\ell) \cdot x(\ell)}{a_p(k)}, \quad x(k) = x(k) + \Delta x(k) \quad \text{C.1}$$

erhöhen, wäre die Helligkeit im Pixel ausreichend. Um den Algorithmus ein wenig sensibler zu machen, erhöht man die betreffende LED nur um einen gewissen Teil. Dies geschieht entweder mit Hilfe des Sensibilitätsfaktors λ oder mit Hilfe des Offsetwertes δ (analog zu Abschnitt 4.7.1). Wenn eine LED erhöht wurde, werden die Helligkeitsdefizite der Pixel neu berechnet. Mit jeder Iteration wird die LED um einen gewissen Anteil erhöht, welche den Pixel mit dem höchsten Defizit dominiert, bis schließlich kein Pixel mehr ein Helligkeitsdefizit aufweist.

Die Vorgehensweise unterscheidet sich von der in Abschnitt 4.7 präsentierten IP dadurch, dass die Prozessierungsreihenfolge der Pixel abhängig vom Bildinhalt ist. Die durch diese Änderung erzielten Ergebnisse sind in Abbildung C-1 zu sehen. Für jeden der drei Prototypen wird die relative Abweichung von der optimalen Lösung im Mittel erheblich reduziert.

Beim Direct-Lit TV kann die relative Abweichung von der optimalen Lösung von 0,84% auf 0,19% und beim Edge-Lit TV von 1,48% auf 0,52% reduziert werden. Die größte Verbesserung wird bei der Signalanzeige erreicht. Hier wird die relative Abweichung von der optimalen Lösung von 3,7% auf 0,61% reduziert.

Die Verwendung der oben beschriebenen Methode hat jedoch zwei Nachteile. Zum einen benötigt man durch das mehrmalige Einlesen der TFT-Werte aus dem RAM und durch die Berechnung der zwischenzeitlich erreichten Helligkeiten in den Pixeln eine zusätzliche Berechnungszeit, zum anderen muss man sich die Defizite der Pixel merken, wodurch ein zusätzlicher Speicheraufwand entsteht.

Beide Nachteile können durch geeignete Verfahren eingedämmt werden. Verwendet man beispielsweise eine Prioritätswarteschlange [Cor 02], werden nur die Pixel neu berechnet, welche zuvor noch nicht die geforderte Helligkeit erreicht haben. Alle Pixel, deren geforderte Helligkeit erreicht ist, fallen aus der Prioritätswarteschlange heraus. Eine geeignete Datenstruktur zum Anlegen einer solchen Prioritätswarteschlange ist der AVL-Baum [Cor 02]. Die Pixel werden in der Prioritätswarteschlange nach deren Helligkeitsdefizit angeordnet. Das Einsortieren der Pixel benötigt logarithmische Zeit zur Gesamtzahl der sich darin befindenden

Pixel. Durch den erhöhten Zeitaufwand ist diese Methode in der Praxis allerdings in zeitkritischen Anwendungen wie Video nicht einsetzbar. Für die Signalanzeige aus Abschnitt 7.4.3 könnte die Methode jedoch aus zwei Gründen in Erwägung gezogen werden: Hier wird meistens kein Video benötigt und die Anzahl der eingeschalteten Pixel ist gering, wodurch ein Einsortieren der Pixel und ein Berechnen der defizitären Pixel weniger Zeit benötigt.

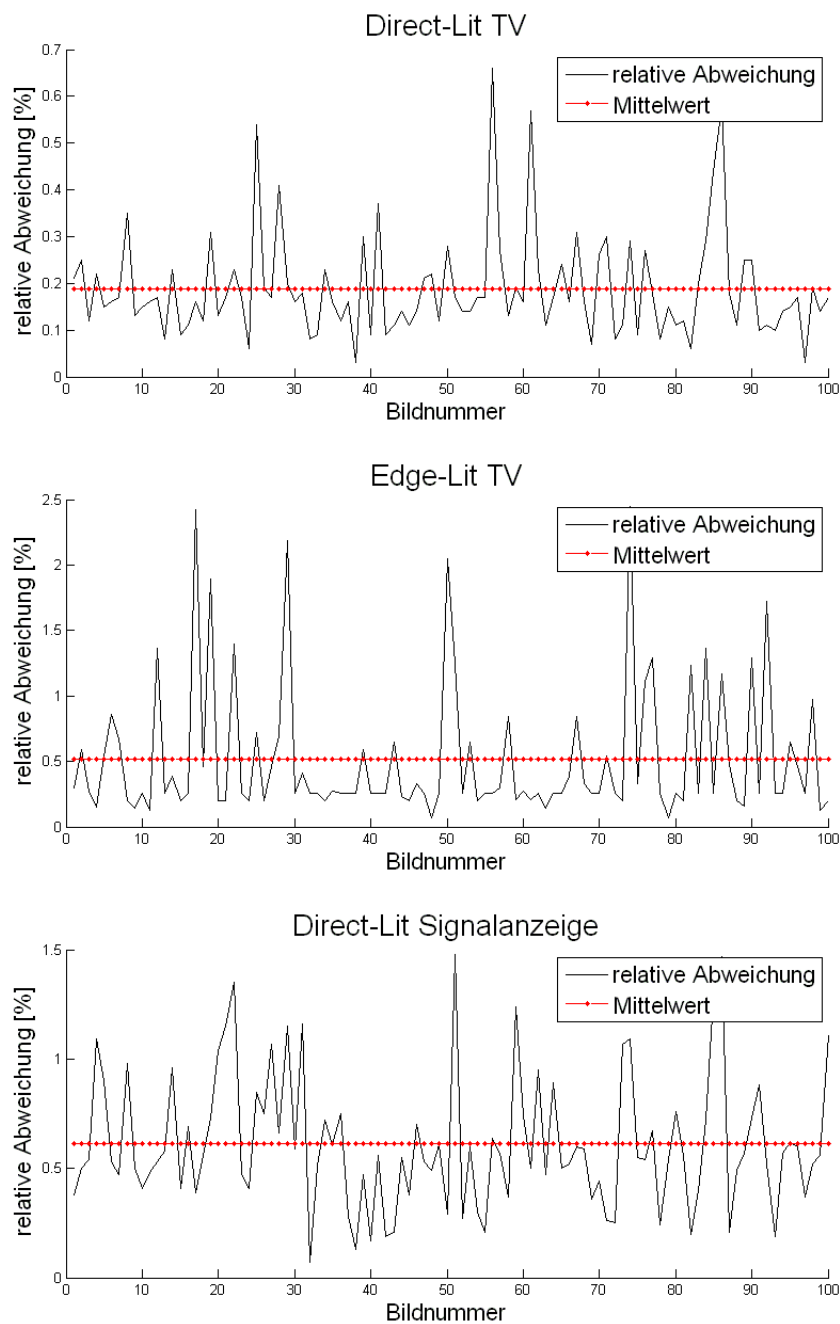


Abbildung C-1: Relative Abweichung von der optimalen Lösung der drei Prototypen bei Verwendung der IP mit bildabhängiger Prozessierereihenfolge der Pixel

Appendix D – Division durch Schiebe- und Vergleichsoperationen

Formel D.1 beschreibt, wie der Wert einer LED k über das bestehende Defizit Δ_p und den Einfluss $a_p(k)$, den die LED im Pixel ausübt, berechnet wird:

$$x(k) = \frac{\Delta_p}{a_p(k)} \quad \text{D.1}$$

Eine Division in Hardware ist sehr komplex und benötigt somit eine große Chipfläche. Darüber hinaus sind manche FPGAs nicht in der Lage, eine Division umzusetzen.

Es wird nun ein einfacher Ansatz der Division durch Schiebe- und Vergleichsoperationen vorgestellt. Die Auflösung des LED-Wertes und des Defizites beträgt 8bit, die Auflösung des Einflusses 10bit.

Das Defizit wird um 10 Stellen nach links verschoben und in ein Register abgelegt. Danach erfolgt die eigentliche Schiebe- und Vergleichsoperation. Diese wird in dem folgenden Pseudocode beschrieben.

```
1  x_temp ← 128
2  b_temp ← 0
3  a_shift ← a << 7
4  b_temp ← b_temp + a_shift
5  a_shift ← a << 6
6  if b_temp <= Defizit
7     then b_temp ← b_temp + a_shift
8         x_temp ← x_temp + 64
9  else
10     then b_temp ← b_temp - a_shift
11         x_temp ← x_temp - 64
12  a_shift ← a << 5
13  if b_temp <= Defizit
14     then b_temp ← b_temp + a_shift
15         x_temp ← x_temp + 32
16  else
17     then b_temp ← b_temp - a_shift
18         x_temp ← x_temp - 32

usw.
```


Zu Beginn wird x_{temp} auf den Wert 128 gesetzt. Der Einfluss wird 7 Stellen nach links verschoben. Dies entspricht einer Multiplikation des Einflusses mit dem Faktor 128. Das Ergebnis der Schiebeoperation wird der aktuell erreichten Helligkeit b_{temp} zugeordnet, welche dann mit dem Defizit im Pixel verglichen werden muss. Falls b_{temp} kleiner als das Defizit ist, dann wird x_{temp} um 64 erhöht und der um 6 Stellen nach links verschobene Einfluss zu b_{temp} hinzuaddiert. Diese Prozedur wird so lange fortgesetzt, bis kein Schieben mehr möglich ist. Wird im letzten Schritt das Ergebnis von b_{temp} bzw. x_{temp} verkleinert, so kann es sein, dass die erreichte Helligkeit nun doch unter der des Defizits liegt. In diesem Fall muss x_{temp} wieder um eins erhöht werden.

Für jede Schiebeoperation benötigt man einen Takt Berechnungszeit. Hinzu kommt ein weiterer Takt für den letzten Schritt der Berechnung. Da das Ergebnis eine Auflösung von 8bit hat, dauert die gesamte Division 9 Takte.

Durch die an den Algorithmus gestellte Bedingung ($b \geq i$) ist dieses Verfahren frei von Fehlern. Bei der Berechnung der TFT-Werte wird das vorgestellte Verfahren leicht modifiziert. Da hier nicht (wie im obigen Beispiel) die LED-Werte aufgerundet werden dürfen, muss man den Abstand zum gewünschten Wert berücksichtigen. Dies geschieht über zwei zusätzliche Vergleiche, welche allerdings keine zusätzliche Zeit benötigen.

Appendix E – Pipelining

Pipelining nutzt die Möglichkeit aus, dass Befehle in einem sequentiellen Befehlsstrom teilweise gleichzeitig ausgeführt werden können. Dadurch wird der Durchsatz einer Schaltung erhöht, ohne den Hardwareaufwand oder die Geschwindigkeit der einzelnen Berechnung zu beeinflussen.

Dies soll am Beispiel der TFT-Berechnung in Abbildung E-1 verdeutlicht werden.

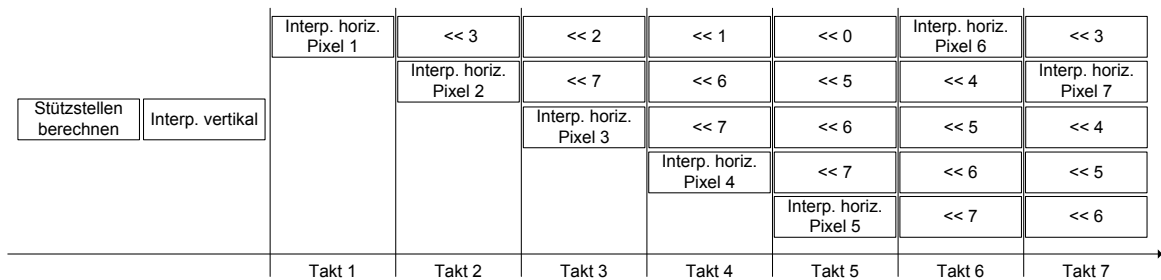


Abbildung E-1: Beispiel für Pipelining

Zunächst werden die Stützstellen berechnet und vertikal interpoliert. Danach erfolgt die horizontale Interpolation. In jedem Takt wird diese für einen Pixel durchgeführt. Dann werden drei Schiebeoperationen pro Pixel ausgeführt, bis schließlich das Ergebnis für den TFT-Wert feststeht. Dieser sequentielle Befehlsstrom benötigt zur Berechnung eines TFT-Wertes 5 Takte. Durch das Pipelining werden die vorhandenen Ressourcen so verteilt, dass diese in jedem Takt verwendet werden. Als Resultat erhält man das Ergebnis für den zweiten Pixel bereits nach dem sechsten Takt, das Ergebnis für den dritten Pixel nach dem siebten Takt usw.

Appendix F – Erzeugen der LSFs

In diesem Appendix werden die verschiedenen Schritte bei der Erzeugung der LSFs beschrieben.

1. Schritt: Erfassen der Helligkeitsverteilung

In einem ersten Schritt muss die Helligkeitsverteilung der LEDs erfasst werden. Hierbei gibt es zwei Möglichkeiten, welche im Folgenden dargelegt werden.

1. Messung der Helligkeitsverteilung

Eine Möglichkeit zur Erfassung der Helligkeitsverteilung der LEDs ist die direkte Messung an einem Display. Die Hauptschwierigkeit besteht darin, dass die Kamera stets unter gleichen Randbedingungen (Umgebunghelligkeit, Abstand, Winkel zum Display etc.) arbeiten muss.

Für den in Abschnitt 7.4.1 präsentierten Prototyp wurde die Helligkeitsverteilung einer einzelnen Leuchtdiode aus den Rohdaten der digitalen Spiegelreflexkamera Nikon D70 gewonnen. Das Ergebnis der Messung wurde, wie in Abbildung F-1 an einem Querschnitt gezeigt, geglättet und schließlich mit Hilfe des Curve-Fitting-Tools von Matlab [Mat 10] gefittet.

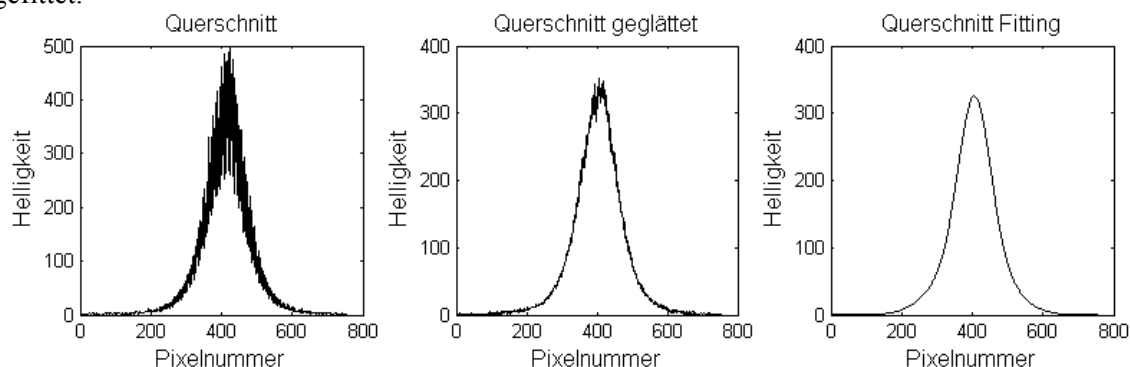


Abbildung F-1: Querschnitt der Helligkeitsverteilung einer LED - links: Messung; Mitte: geglättete Messung; rechts: Fitting der Messung

Aus dem Ergebnis des Fittings wurde das Modell einer Leuchtdiode erstellt (siehe Abbildung F-2). Jede der 180 LEDs des Backlights besteht aus 12 (4·3) dieser in einer Gitterstruktur angeordneten Leuchtdioden.

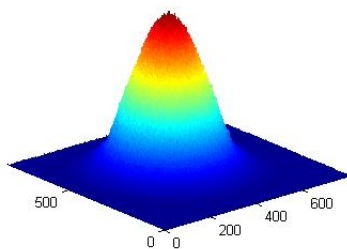


Abbildung F-2: Modell einer Leuchtdiode

Die Helligkeitsverteilung der LED, die aus 12 dieser Leuchtdioden besteht und als Vorlage für alle 180 LEDs des Displays diente, wurde durch Überlagerung der zwölf Helligkeitsverteilungen ermittelt.

Ebenfalls aus den Rohdaten der digitalen Spiegelreflexkamera Nikon D70 wurde die Helligkeitsverteilung der sechs LEDs des in Abschnitt 7.4.2 vorgestellten Prototyps eines Edge-Lit TVs gewonnen.

2. Physikalische Simulation der Helligkeitsverteilung

Man kann die Ausbreitung von Licht in optomechanischen Systemen mit Hilfe spezieller Software, wie beispielsweise SPEOS [SPE 10], ASAP [ASA 10] oder TracePro [Tra 10], physikalisch simulieren. Dadurch erhält man unter anderem die benötigten Ergebnisse in Bezug auf die Verteilung des Lichtes an der Oberfläche des Displays. Um aussagekräftige Ergebnisse zu erhalten, müssen exakte Modelle der LEDs und der Umgebung generiert werden.

Die Helligkeitsverteilung der LEDs des in Abschnitt 7.4.3 vorgestellten Prototyps wurde mit der Software SPEOS ermittelt.

2. Schritt: Skalierung der Pixelauflösung

Mit den beiden oben gezeigten Verfahren erhält man die Helligkeitsverteilung der LEDs. Die Auflösung dieser stimmt im Allgemeinen nicht mit der des Displays überein. Daher muss die Auflösung, wie in Abbildung F-3 an einem Beispiel gezeigt, skaliert werden.



Abbildung F-3: links: simulierte Helligkeitsverteilung einer LED; rechts: skalierte Helligkeitsverteilung einer LED

In dem gezeigten Beispiel wurden die simulierten Werte in der Auflösung von (76·171) Pixeln auf die benötigten (104·576) Pixel skaliert. Für die Skalierung kann ein Interpolationsverfahren angewandt werden. In dieser Arbeit wurde die „imresize“-Funktion von Matlab [Mat 10] zur Skalierung der Pixelauflösung verwendet.

3. Schritt: Formatierung des Einflusses

Nachdem die Helligkeitsverteilungen der LEDs ermittelt und auf die benötigte Auflösung skaliert wurden, werden diese additiv überlagert. Die linke Seite der Abbildung F-4 zeigt eine solche Überlagerung.

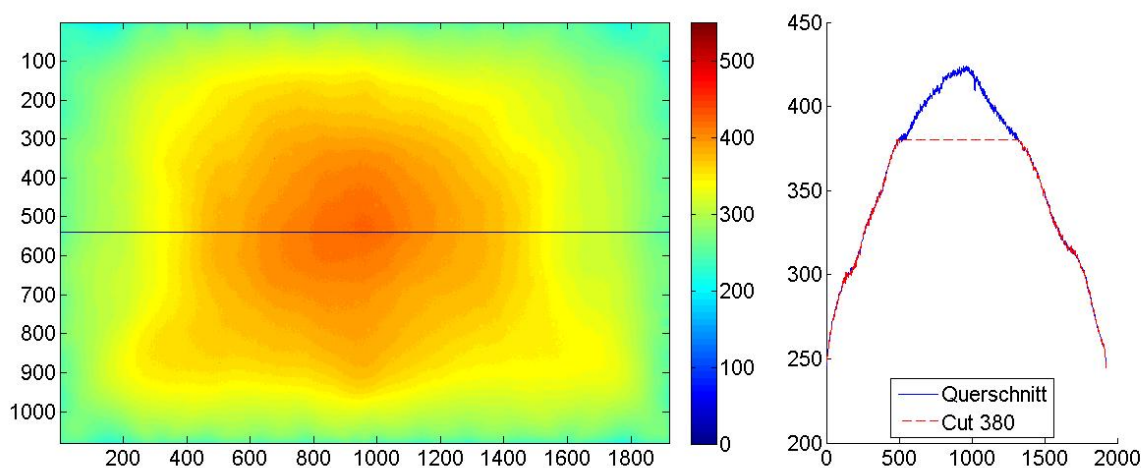


Abbildung F-4: links: Überlagerung der Helligkeitsverteilungen; rechts: Querschnitt und Schnitt bei 380cd/m²

Die Skala der Überlagerung gibt die Helligkeit des Displays in cd/m² an. Für gewöhnlich ist die Helligkeit in der Mitte des Displays am größten und fällt zu den Rändern hin mehr oder weniger stark ab.

Laut Spezifikation soll die erreichte Helligkeit in der Mitte des Displays bei b^{spec} liegen. Normalerweise ergibt sich dann in der Mitte ein Überschuss an Helligkeit, welcher abgeschnitten werden kann (siehe rechte Seite der Abbildung F-4; b^{spec} ist hier 380cd/m²). Die Überlagerung ist Grundlage für die Formatierung der Einflüsse der LSFs. Für jeden Pixel, dessen Helligkeit unter b^{spec} liegt, muss gelten, dass die Summe der Einflüsse aller LEDs gleich eins ist:

$$b^{\leq} = \left\{ b_p : b_p \leq b^{spec} \cap \sum_{\ell=1}^L a_p(\ell) = 1 \right\} \quad \text{F.1}$$

Der Einfluss einer LED k auf den Pixel p ergibt sich dann aus dem Quotienten der erreichten Helligkeit und dem nicht normierten Einfluss d_p :

$$a_p(k) = \frac{d_p}{b_p} \quad \text{F.2}$$

Für die übrigen Pixel, deren Helligkeit größer als b^{spec} ist, ist die Summe aller Einflüsse größer als eins.

$$b^{>} = \left\{ b_p : b_p > b^{spec} \cap \sum_{\ell=1}^L a_p(\ell) = \frac{b_p}{b^{spec}} \right\} \quad \text{F.3}$$

Der Einfluss einer LED k auf den Pixel p ergibt sich aus der erreichten Helligkeit im Pixel b_p , dem nicht normierten Einfluss d_p und der spezifizierten Helligkeit b^{spec} :

$$a_p(k) = \frac{d_p}{b_p} \cdot \frac{b_p}{b^{spec}} \quad \text{F.4}$$

Auf diese Art erhält man die LSF für jede LED.

4. Schritt: Kondensierung der LSFs für SSC und Kompression der LSFs für die Postprozessierung

Die ermittelten LSFs werden (wie weiter oben beschrieben) zur Berechnung von SSC und für die Postprozessierung benötigt.

Kondensierung der LSFs für SSC

Analog zu Abschnitt 5.3 wird die maximale Zellgröße ermittelt. Die kondensierten LSFs werden dann über den Mittelwert der Zelle berechnet und im LSF-RAM für SSC abgelegt.

Kompression der LSFs für Postprozessierung

Für die in Kapitel 1 vorgestellte Postprozessierung muss die erreichte Helligkeit in jedem Pixel berechnet und der Überschuss an Helligkeit von der TFT-Zelle herausgefiltert werden. Um Ressourcen zu sparen, wird auch hier nur ein Bruchteil der Information der LSFs im Speicher hinterlegt. Die Stützstellen sind in Form eines Gitters angeordnet. Der Abstand der Stützstellen beträgt S' in horizontaler und T' in vertikaler Richtung. Zwischen diesen Stützstellen wird die Helligkeit in allen Pixeln linear interpoliert (siehe auch Abbildung 7-4).

5. Schritt: Boosten der LEDs

Im Allgemeinen gilt, je geringer der Einfluss der dominierenden LED auf einen hellen Pixel ist, desto schlechter ist dies für die mögliche Leistungersparnis, da mehrere LEDs auf einen hohen Wert gesetzt werden müssen.

Aus diesem Grund sieht man für Displays mit Local Dimming vor, die LEDs zu boosten. Dadurch ist man in der Lage, die Helligkeit lokaler zu erzeugen und somit das Backlight besser an die Eigenschaften des Bildes anpassen zu können. Je höher der Boost-Faktor, desto höher ist die erreichbare Leistungersparnis, wobei ab einem gewissen Schwellwert eine Sättigung eintritt.

Dem Boost-Faktor sind jedoch physikalische Grenzen gesetzt. Der Strom, der durch die LED fließt, kann nicht beliebig erhöht werden. Darüber hinaus verschlechtert sich der Wirkungsgrad der LEDs mit zunehmendem Strom (dies wurde in Abbildung F-5 nicht berücksichtigt). Auch ist die Zeit, in der die LED leuchtet, Einschränkungen unterworfen.

Daneben gibt es auch wirtschaftliche Grenzen. Je heller eine LED im Allgemeinen ist, desto teurer ist sie.

Bei der Festlegung des Boost-Faktors müssen selbstverständlich die physikalischen und wirtschaftlichen Faktoren berücksichtigt werden. Für jedes Display muss ein Kompromiss aus Mehrkosten für das Backlight und möglicher Leistungersparnis geschlossen werden. Ein Boost-Faktor von 1,3 hat sich als guter Kompromiss herausgestellt.

Die Einträge der LSFs müssen beim Boosten der LEDs an den entsprechenden Boost-Faktor angepasst werden.

Formelzeichen, Variablen, Abkürzungen und Glossar

Formelzeichen

| | |
|----------------|--|
| A | Einflussmatrix. Beschreibt den Einfluss aller LEDs auf alle Pixel des Bildes. Die Anzahl der Zeilen entspricht der Anzahl der Pixel, die Anzahl der Spalten entspricht der Anzahl der LEDs. |
| $a_p(k)$ | Einfluss der k -ten LED auf den Pixel p . |
| $a_p(q)$ | Einfluss der LEDs q auf den Pixel p . |
| $a_{p,k}$ | Einfluss der k -ten LED auf den Pixel p (Matrixschreibweise). |
| α | Verhältnis aus erreichter Helligkeit und Sollhelligkeit beim Clipping. |
| b | Helligkeit des Backlights in allen Pixeln. Hierbei handelt es sich um einen Spaltenvektor. Die Anzahl der Einträge entspricht der Anzahl der Pixel. |
| b_p | Helligkeit eines Pixels p . |
| b^{\leq} | Menge aller Pixel, deren Helligkeit bei eingeschaltetem Backlight unter der spezifizierten Helligkeit liegt oder gleich der spezifizierten Helligkeit ist. |
| $b^{>}$ | Menge aller Pixel, deren Helligkeit bei eingeschaltetem Backlight über der spezifizierten Helligkeit liegt. |
| b^{spec} | Spezifizierte Helligkeit. Wegen geringer Homogenität des Backlights wird diese meist nur in den Pixeln in der Mitte des Displays erreicht. |
| b_p^{All-On} | Helligkeit von Pixel p bei voller Ausleuchtung des Backlights. |
| c_p | Clipping im Pixel p , umgerechnet in einen Grauton. Die Auflösung in bit entspricht der der TFT-Werte. |
| Γ | Gamma-Korrektur-Wert. Dieser liegt bei kommerziellen TVs bei etwa 2,2. |
| γ | Verhältnis zwischen der Sollhelligkeit im Pixel und der erreichten Helligkeit im Pixel; wird bei der Postprozessierung bei nichtlinearem Verhältnis zwischen TFT-Wert und Helligkeit benötigt. |
| $d_p(k)$ | Nichtnormierter Einfluss der k -ten LED auf den Pixel p . |

| | |
|---------------------|--|
| Δ_b | Helligkeit, die von der geforderten Helligkeit im Pixel subtrahiert wird; wird beim Prototyp der Signalanzeige zur Erhöhung der Leistungersparnis verwendet. |
| Δ_p | Helligkeitsdefizit in Pixel p . |
| $\Delta x(k)$ | Der Anteil, um den der LED-Wert $x(k)$ erhöht werden muss, um die geforderte Helligkeit in einem Pixel zu erreichen. |
| $\Delta x(q)$ | Die Anteile, um den die LEDs $x(q)$ jeweils erhöht werden müssen, um die geforderte Helligkeit in einem Pixel zu erreichen. |
| δ | Offset-Wert. Dieser wird in der IP verwendet. Mit steigender Iterationszahl wird der Offset-Wert kleiner. |
| λ | Sensibilitätsfaktor. Dieser wird in der Intermediate Phase verwendet. Mit steigender Iterationszahl wird der Sensibilitätsfaktor größer. |
| i | Bilddaten (geforderte Helligkeit). Hierbei handelt es sich um einen Spaltenvektor. Die Anzahl der Einträge entspricht der Anzahl der Pixel. |
| i_p | Geforderte Helligkeit in Pixel p . |
| i_p^{\max} | Die in einem Pixel p erreichbare Helligkeit bei ungedimmtem Backlight und maximalem TFT-Wert. |
| i_p^{soft} | Geforderte Helligkeit in Pixel p nach Soft-Clipping. |
| i'_p | Geforderte Helligkeit im kondensierten Pixel p' . |
| P_v | Verlustleistung des Backlights; proportional zur Summe aller PWM-Werte der LEDs. |
| t_p | TFT-Wert von Pixel p . |
| x | LED-Werte. Hierbei handelt es sich um einen Spaltenvektor. Die Anzahl der Einträge entspricht der Anzahl der LEDs. |
| $x(k)$ | PWM-Wert der k -ten LED. |
| x_k | PWM-Wert der k -ten LED (Matrixschreibweise). |
| $x_{pre}(k)$ | PWM-Wert der k -ten LED des Presets der LB. |

- $x(q)$ PWM-Wert der LEDs, die eine Zelle dominieren.
- $x'(k)$ Intensität, mit der die k-te LED leuchtet.

Variablen

- f Bildnummer bei Videoanwendungen.
- i Laufvariable für Pixel eines Displays in y-Richtung.
- i' Laufvariable für kondensierte Pixel eines Displays in y-Richtung.
- j Laufvariable für Pixel eines Displays in x-Richtung.
- j' Laufvariable für kondensierte Pixel eines Displays in x-Richtung.
- k Nummer der aktuell betrachteten LED.
- L Anzahl der LEDs.
- ℓ Laufvariable für LEDs.
- M Anzahl der Spalten eines Displays.
- M' Anzahl der kondensierten Spalten eines Displays.
- N Anzahl der Zeilen eines Displays.
- N' Anzahl der kondensierten Zeilen eines Displays.
- n Anzahl der Iterationen der IP. Die n-te Iteration entspricht den FDC.
- P Anzahl der Pixel eines Displays. Jeder Pixel ist im Allgemeinen unterteilt in drei Subpixel in den Farben Rot, Grün und Blau.
- P' Anzahl der kondensierten Pixel eines Displays.
- p Laufvariable für die Pixel eines Displays, wenn diese in Spaltenform angeordnet sind.
- p' Laufvariable für die kondensierten Pixel eines Displays, wenn diese in Spaltenform angeordnet sind.
- S Anzahl der Spalten bei der Kondensierung.

| | |
|------------|---|
| s | Anzahl der Pixel einer kondensierten Zelle in x-Richtung. |
| S' | Anzahl der interpolierten Spalten bei der Postprozessierung. |
| S^{soft} | Anzahl der Spalten bei der Postprozessierung mit Soft-Clipping. |
| T | Anzahl der Zeilen bei der Kondensierung. |
| t | Anzahl der Pixel einer kondensierten Zelle in y-Richtung. |
| T' | Anzahl der interpolierten Zeilen bei Postprozessierung. |
| T^{soft} | Anzahl der Zeilen bei der Postprozessierung mit Soft-Clipping. |

Abkürzungen und Glossar

| | |
|-------------------------------------|---|
| Approximations- algorithmus | Algorithmus, mit dem ein Optimierungsproblem näherungsweise, fast optimal, gelöst werden kann. |
| ASIC/ASSP | Application Specified Integrated Circuit bzw. anwendungsspezifisches Standardprodukt: Integrierter Schaltkreis, der in seiner Funktion nicht mehr veränderbar ist. |
| Backlight | Hintergrundbeleuchtung eines LCDs. |
| Bilddaten (adaptierte Bilddaten) | Bestehen aus den TFT-Werten für alle Pixel. In der Postprozessierung werden die Bilddaten an die Helligkeit des gedimmten Backlights angepasst. Das Resultat wird als adaptierte Bilddaten oder als adaptierte TFT-Werte bezeichnet. |
| Black-Extended-Modus | Surcondenser-Modus. Die Bilddaten werden um einen schwarzen Rahmen erweitert. Das mit dem Condenser prozessierte Bild wird dadurch an den Rändern abgedunkelt. |
| CAF | Clipping Allowance Factor des automatischen Condenser-Modus, der sich aus dem maximalen Pixelwert einer Zelle und dem arithmetischen Mittelwert der Pixel einer Zelle errechnet. |
| Clipping | Ein Phänomen, welches auftritt, wenn die vom Backlight erzeugte Helligkeit in einem Pixel geringer ist als die vom Bild geforderte Helligkeit in diesem Pixel. Clipping führt zu einem Informationsverlust beim dargestellten Bild und kann zu einer Verschlechterung der Bildqualität führen. Daher gilt es in der |

| | |
|--|---|
| | Regel, Clipping zu vermeiden. |
| Condenser | Der Condenser kondensiert eine Zelle von $(s \cdot t)$ Pixeln zu einem kondensierten Pixel. Hierbei handelt es sich um den Präprozessor von SSC zur Verringerung der Komplexität bei gleichzeitiger Erhöhung der Leistungersparnis. |
| Condenser-Modus | Es wird unterschieden zwischen starren Condenser-Modi, welche über alle Zellen die gleiche Kondensierungsfunktion verwenden und automatischem Condenser-Modus, welcher mehrere Kondensierungsfunktionen verwendet. |
| Crosstalk | Übersprechen der Helligkeit der LEDs. Eine LED beeinflusst auch Pixel, die von anderen LEDs dominiert werden. |
| Cut-Off-Modus | Surcondenser-Modus. Surcondenser-Zelle ragt nicht über den eigentlichen Bildbereich heraus. |
| Direct-Lit Backlight (direktes Backlight) | Lichtquellen sind hinter dem Display angebracht. |
| Edge-Lit Backlight (indirektes Backlight) | Wird auch Side-Lit Backlight genannt. Die Lichtquellen sind an den Seiten des Displays angebracht. |
| Einflussmatrix | Beschreibt den Einfluss aller LEDs auf alle Pixel des Bildes. Die Anzahl der Zeilen entspricht der Anzahl der Pixel, die Anzahl der Spalten entspricht der Anzahl der LEDs. |
| Farb-Crosstalk | Phänomen bei 3D-Dimming. Eine farbige LED beeinflusst nicht nur die Helligkeit im Subpixel der eigenen Farbe, sondern auch in den der anderen Farben. |
| FIFO | First In - First Out. Eine Warteschlange, die Daten in der Reihenfolge ihres Einfügens ausgibt. |
| Final Duty Cycles (FDC) | Final Duty Cycles. Letzte Phase des SSC-Algorithmus. Endgültige Zuweisung der LED-Werte. Entspricht bei Verwendung der IP deren letzten Phase. |
| Flicker | Starke Helligkeitsschwankungen einer oder mehrerer LEDs von einem Frame zu einem oder mehreren anderen, welche vom Betrachter als störend wahrgenommen werden. |

| | |
|---------------------------------|---|
| FPGA | Field Programmable Gate Array. Integrierter Schaltkreis, in den eine logische Schaltung programmiert werden kann. Die Prototypen in dieser Arbeit werden mit Verilog und VHDL erzeugt. |
| Full-HD | (1920·1080·3) Pixel; gängige Auflösung für LCD-TVs. |
| Frame | Ein einzelnes Bild aus einer Videosequenz. |
| Grauton | Bei einer Kodierung der TFT-Werte von g -bit gibt es insgesamt 2^g verschiedene Grautöne. |
| Helligkeit | In der Displaytechnik wird üblicherweise für die Helligkeit eines Pixels der fotometrische Begriff Luminanz verwendet. In dieser Arbeit geht es jedoch darum, eine gegebene Eingangsgröße (TFT-Wert) durch Vergleichsoperationen in eine Ausgangsgröße (einen adaptierten TFT-Wert) umzuwandeln. Der physikalische Charakter der Größe spielt dabei keine Rolle. Daher wird in dieser Arbeit der Begriff Helligkeit zum besseren Verständnis verwendet. |
| Helligkeitsverteilung einer LED | Nicht normierter Einfluss einer LED auf alle Pixel. |
| Intermediate Phase (IP) | Intermediate Phase. Optionale Zwischenphase des SSC-Algorithmus. Besteht aus n Iterationen. Die LED-Werte werden sukzessive an das darzustellende Bild angepasst. |
| Kondensierungsfunktion | Mathematische Vorschrift, wie der kondensierte Pixel aus einer Zelle von Pixeln bestimmt werden soll. Einem starren Condenser-Modus wird eine Kondensierungsfunktion zugewiesen. Der automatische Condenser-Modus verwendet verschiedene Kondensierungsfunktionen. |
| Lower Bounds (LB) | Lower Bounds. Erste Phase des SSC-Algorithmus. Als Ergebnis erhält man die bildabhängige mathematische untere Schranke der LED-Werte. |
| LED | Wenn in dieser Arbeit der Begriff LED verwendet wird, so bezeichnet er, wenn nicht anders angegeben, eine Reihenschaltung von einer oder mehreren Leuchtdioden. |
| Lichtverluste (Light Leakage) | Trotz einer Transmission von 0 dringt noch ein Rest Helligkeit des Backlights durch die LC-Zelle. Dadurch erscheint ein eigentlich schwarzer Pixel in einem dunklen Grau, wodurch das statische |

| | |
|---------------------------------|---|
| | Kontrastverhältnis reduziert wird. |
| Lichtverteilungsfunktion | siehe LSF. |
| Luminanz | Fotometrisches Maß für die Helligkeit von Pixeln [cd/m^2]. |
| LUT | Look-Up-Table, zu Deutsch: Nachschlagetabelle. Zuvor errechnete Werte werden in dieser Tabelle hinterlegt, um sich eine oftmals aufwändige Berechnung zu ersparen. |
| LSF (Light Spread Function) | Zu Deutsch: Lichtverteilungsfunktion. Es handelt sich um den normierten Einfluss einer LED auf alle Pixel. |
| Optimal Global Dimming (OGD) | Dieser Wert dient als Referenz zur eingesparten Leistung. Optimal Global Dimming entspricht der optimalen Lösung für ein Bild, in dem jeder Pixel die maximale Helligkeit aufweist. |
| Partition | Bei der Berechnung der Final Duty Cycles wird das Bild in Partitionen zerlegt. Innerhalb der Partition werden die Pixel nach einem vorgegebenen Muster abgearbeitet und die die Partition begrenzenden LEDs gegebenenfalls geupdatet. |
| Pixel Distortion | Siehe Clipping. |
| Sektor | Die Berechnung der IP und der FDC kann durch Parallelisieren beschleunigt werden. Das Bild wird dabei in verschiedene Bereiche, genannt Sektoren, aufgeteilt. Parallel abzuarbeitende Sektoren sollten dabei voneinander möglichst weit räumlich getrennt werden. |
| Surcondenser | Kondensiert ($s \cdot t$) Pixel zu einem Surcondenser Pixel. Die Zelle eines Surcondensers überlappt die des Condensers; wird verwendet, um noch höhere Leistungersparnis zu erreichen. |
| TFT-Wert (adaptierter TFT-Wert) | Diskreter Wert (üblicherweise zwischen 8bit und 10bit Auflösung). Wird im Spaltentreiber in eine analoge Spannung umgewandelt. Jedem Supixel des Bildes wird ein TFT-Wert zugewiesen. In der Postprozessierung werden die TFT-Werte auf die Helligkeit des gedimmten Backlights angepasst. Das Resultat wird als adaptierter TFT-Wert bezeichnet. |
| Zelle | Beinhaltet ($s \cdot t$) Pixel, welche zu einem Pixel kondensiert werden. |

Literaturverzeichnis

- [Alt 10] <http://www.altera.com>, zugegriffen am 26.09.2010
- [An 08] **An, J.-Y./ Kim, S.-E./ Song, W.-J./ Lee, T.-W./ Kim, C.-G. (2008):** *Adaptive Local Dimming Backlight for Liquid-Crystal Displays*. In: Proceedings of IDW 2008: S. 285-288
- [Ana 08] **Anandan, Munisamy (2008):** *Progress of LED backlights for LCDs*. In: Journal of SID 2008: S. 287-310
- [ASA 10] <http://www.breault.com/software/asap.php>, zugegriffen am 26.09.2010
- [BT 97] **Bertsimas, Dimitris/ Tsitsiklis, John N. (1997):** *Introduction to Linear Optimization*. Athena Scientific Series in Optimization and Neural Computation
- [Ch 06] **Chen, Hanfeng (2006):** *Backlight Local Dimming Algorithm for High Contrast LCD-TV*. In: Proceedings of ASID 2006: S. 168-171
- [Ch 07] **Chen, Hanfeng (2007):** *Locally pixel-compensated backlight dimming on LED-backlit LCD TV*. In: Journal of SID 2007: S. 981-988
- [Ch 10] **Chen, Hanfeng/ Ha, Tae Hyeun/ Sung, Jun Ho/ Kim, Hyung Rae/ Han, Baik Hee/ Ahn, Yoon Soon (2010):** *Quantified Evaluation for Clipping Artifact of Local Dimming LCDs*. In: Proceedings of SID 2010: S. 315-318
- [Che 05] **Cheng, Wayne (2005):** *Power Minimization of LED Backlight in a Color Sequential Display*. In: Proceedings of SID 2005: S.1384-1387
- [CKK 08] **Cho, H.-S./ Kim, H./ Kwon, O.-K. (2008):** *Adaptive Selection Algorithm for LED Backlight of LCD-TVs*. In: Proceedings of IDW 2008: S. 281-284
- [Co 02] **Cormen, Thomas H./ Leiserson, Charles E./ Rivest, Ronald L./ Stein, Clifford (2002):** *Algorithmen – Eine Einführung*. 2. Aufl., McGraw-Hill Higher Education
- [Da 98] **Dantzig, George B. (1966):** *Lineare Programmierung und Erweiterungen*. Berlin, Heidelberg, New York: Springer Verlag
- [Fer 70] **Ferguson, James L. (1970):** *Display devices utilizing liquid crystal light modification*. Patentschrift US3731986

- [GH 07] **Greef, Pierre de/ Hulze, Hendriek Groot (2007):** *Adaptive Dimming and Boosting Backlight for LCD-TV Systems*. In: Proceedings of SID 2007: S. 1332-1335
- [GW 08] **Gonzales, Rafael/ Woods, Richard (2008):** *Digital Image Processing*. Pearson International
- [Hei 02] **Heinkel, Ulrich/ Padeffke, Martin/ Haas, Werner/ Buerner, Thomas/ Braisz, Herbert/ Gentner, Thomas/ Grassman, Alexander (2002):** *The VHDL Reference – A Practical Guide to Computer-Aided Integrated Circuit Designs*. New York: John Wiley & Sons
- [HG 09] **Greef, Pierre de/ Hulze, Hendriek Groot (2009):** *Power Savings by Local Dimming on a LCD Panel with Side Lit Backlight*. In: Proceedings of SID 2009: S. 749-752
- [HLL 08] **Hsu, Ching-Fu/ Lai, Chih-Chang/ Li, Jyun-Sian (2008):** *Backlight Power Reduction and Image Contrast Enhancement using Adaptive Dimming for Global Backlight Applications*. In: Proceedings of SID 2008: S. 776-779
- [HMS 97] **Hering, Ekbert/ Martin, Rolf/ Stohrer, Martin (1997):** *Physik für Ingenieure*. 6. Aufl., Springer Verlag
- [Hop 06] **Hoppe, Bernhard (2006):** *Verilog – Modellbildung für Synthese und Verifikation*. Oldenbourg Verlag
- [HS 70] **Helfrich, Wolfgang/ Schadt, Martin (1970):** *Lichtsteuerzelle*. Patentschrift CH532261
- [Hsi 08] **Hsieh, Chun-Yu/ Wang, Shih-Jung/ Lee, Yu-Huei/ Chen, Ke-Horng/ Ho, Ming-Tsung (2008):** *High Efficiency and/or Low Cost LED Backlight System for Color Sequential*. In: Proceedings of SID 2008: S. 1104-1107
- [IEC 09] **TV Power Consumption Standard, IEC 62087-BD Edition 2.0 (2008):** http://webstore.iec.ch/preview/info_iec62087-BD{ed2.0}en.pdf, zugegriffen am 26.09.2010
- [ILOG 10] <http://www-01.ibm.com/software/integration/optimization/cplex-optimizer>, zugegriffen am 26.09.2010
- [Kob 09] **Kobayashi, Shunsuke/ Mikoshiba, Shigeo/ Lim, Sungkyoo (2009):** *LCD Backlights*. New York: Wiley-SID Series in Display Technology, John Wiley & Sons

- [KS 00] **Kories, Ralf/ Schmidt-Walter, Heinz (2000):** *Taschenbuch der Elektrotechnik*. 4. überarbeitete Aufl., Verlag Harri Deutsch
- [Lee 08] **Lee, Jiun-Haw/ Liu, David N./ Wu, Shin-Tson (2008):** *Introduction to Flat Panel Displays*. New York: Wiley-SID Series in Display Technology, John Wiley & Sons
- [Lin 09] **Lin, Fang-Cheng/ Huang, Yi-Pai/ Wie, Chang-Ming/ Shieh, Han-Ping (2009):** *Color-breakup suppression and low-power consumption by using the Stencil-FSC method in field-sequential LCDs*. In: Journal of SID 2009: S. 221-228
- [Lu 01] **Lueder, Ernst (2001):** *Liquid Crystal Displays*. New York: Wiley SID-Series in Display Technology, John Wiley & Sons
- [Lu 08] **Lu, Ruibo/ Nie, Xiangyi/ Wu, Shin-Tson (2008):** *Color performance of an MVA-LCD using an LED backlight*. In: Journal of SID 2008: S. 1139-1145
- [Mat 10] <http://www.mathworks.de>, zugegriffen am 26.09.2010
- [Nik 10] <http://www.digitalkamera.de/Kamera/Nikon/D70s.aspx>, zugegriffen am 26.09.2010
- [Pen 07] **Peng, Huajun J./ Zhang, Wei/ Hung, Chun-Kit/ Tsai, Chen-Jung/ Ng, Kwan-Wah/ Chen, Shou-Lung/ Huang, Danding/ Chueng, Yuk-Lung/ Liu, Ying (2007):** *High Contrast LCD TV using Active Dynamic LED Backlight*. In: Proceedings of SID 2007: S. 1336-1339
- [Poy 02] **Poynton, Charles (2002):** *Digital Video and HDTV – Algorithms and Interfaces*. Morgan Kaufmann
- [Sa 09] http://downloadcenter.samsung.com/content/UM/200905/20090520185627656/BN68-02328D-00L04_0515.pdf, zugegriffen am 26.09.2010
- [Se 04] **Seetzen, Helge/ Heidrich, Wolfgang/ Stuerzlinger, Wolfgang/ Ward, Greg/ Whitehead, Lorne/ Trentacoste, Matthew/ Ghosh, Abhijeet/ Vorozcovs, Andrejs (2004):** *High Dynamic Range Display Systems*. In: ACM SIGGRAPH 2004: S. 760-768
- [Sh 08] **Shiga, T. (2008):** *Power Reduction of LCTVs with an Adaptive Dimming Technique for Standard TV Programs*. In: Proceedings of IDW 2008: S. 747-750

- [Shi 07] **Shirai, T./ Shimizukawa, S./ Shiga, T./ Mikoshiba, S. (2007):** *RGB-LED Backlights for LCD-TVs with 0D, 1D and 2D Adaptive Dimming.* In: Proceedings of SID 2007: S. 1520-1523
- [SPE 10] <http://www.optis-world.com>, zugegriffen am 26.09.2010
- [Tra 10] <http://www.lambdares.com>, zugegriffen am 26.09.2010
- [Xu 08] **Xu, Chihao/ Karrenbauer, Andreas/ Soh, Kian Min/ Codrea, Cosmin (2008):** *Consecutive multiline addressing: A scheme for addressing PMOLEDs.* In: Journal of SID 2008: S. 211-219
- [Yam 02] **Yamada, Fumiaki/ Nakamura, Hajime/ Sakaguchi, Yoshitami/ Taira, Yoichi (2002):** *Sequential-color LCD Based on OCB with an LED Backlight.* In: Journal of SID 2002: S. 81-85
- [Yeo 08] **Yeo, Dong-Min/ Kwon, Yong-Hoon/ Kang, Eui-Jeong/ Park, Se-Ki/ Yang, Byungchoon/ Kim, Gicheol/ Jang, Taeseok (2008):** *Smart Algorithms for Local Dimming LED Backlight.* In: Proceedings of SID 2008: S. 986-989