



**Processing Decoded Video for  
Backlight Dimming**  
Video Quality Enhancement  
on LCD with Dynamic Local Backlight

**Nino Burini**  
**PhD Thesis**  
**March 2013**

# Processing Decoded Video for Backlight Dimming

Video Quality Enhancement on LCD with Dynamic Local Backlight

Nino Burini

March 2013

Technical University of Denmark



**DTU Fotonik**

Department of Photonics Engineering

---

Coding and Visual Communication Group  
DTU Fotonik  
Technical University of Denmark  
Ørsteds Plads, Building 343  
2800 Kongens Lyngby  
DENMARK

## Eldorado

Gaily bedight,  
A gallant knight,  
In sunshine and in shadow,  
Had journeyed long,  
Singing a song,  
In search of Eldorado.

But he grew old -  
This knight so bold -  
And o'er his heart a shadow  
Fell as he found  
No spot of ground  
That looked like Eldorado.

And, as his strength  
Failed him at length,  
He met a pilgrim shadow -  
"Shadow," said he,  
"Where can it be -  
This land of Eldorado?"

"Over the Mountains  
Of the Moon,  
Down the Valley of the Shadow,  
Ride, boldly ride,"  
The shade replied -  
"If you seek for Eldorado!"

# Abstract

Quality of digital image and video signals on TV screens is affected by many factors, including the display technology and compression standards. An accurate knowledge of the characteristics of the display and of the video signals can be used to develop advanced algorithms that improve the visual rendition of the signals, particularly in the case of LCDs with dynamic local backlight. This thesis shows that it is possible to model LCDs with dynamic backlight to design algorithms that improve the visual quality of 2D and 3D content, and that digital video coding artifacts like blocking or ringing can be reduced with post-processing.

LCD screens with dynamic local backlight are modeled in their main aspects, like pixel luminance, light diffusion and light perception. Following the model, novel algorithms based on optimization are presented and extended, then reduced in complexity, to produce backlights that find optimal balance between image quality and power consumption. The algorithms are tested in several experiments in comparison with other State of the Art approaches.

Based on the model of backlight dimming, another model of backlight scanning for crosstalk reduction in time-sequential stereoscopic visualization on LCD is introduced. Crosstalk at a given luminance level is minimized by properly adjusting the signals controlling the backlight.

# Resumé

Kvaliteten af digitale billed- og videosignaler på TV skærme er påvirket af mange faktorer, herunder display-teknologi og kompressions-standard. Præcist kendskab til skærmens og videosignalernes egenskaber kan bruges til at udvikle avancerede algoritmer til at forbedre den visuelle gengivelse af signalerne, særligt i tilfælde af LCD med dynamisk lokal bagbelysning. Denne afhandling viser, at det er muligt at modellere LCD med dynamisk baglys til at udvikle algoritmer, der forbedrer den visuelle kvalitet af 2D og 3D billede, og at videokodnings-artefakter som blokke eller ringinger kan reduceres ved hjælp af post-processing.

LCD-skærme med dynamisk lokal bagbelysning modelleres med hensyn til deres vigtigste aspekter, såsom pixel luminans, lysdiffusion og perception. Nye algoritmer baseret på optimering efter modellen præsenteres, udvides og simplificeres for at producere bagbelysning, der har en optimal balance mellem billedkvalitet og strømforbrug. Algoritmerne testes i flere eksperimenter i sammenligning med andre avancerede metoder.

En anden model af bagbelyst scanning indføres for at reducere cross-talk i tid-sekventiel stereoskopisk visualisering på LCD skærme. Cross-talk på et givet luminansniveau minimeres ved at justere de signaler, der styrer bagbelysningen.

# Acknowledgements

This Ph. D. thesis is the final product of a three year long project where I have dedicated all of my efforts to save the World with LCD local backlight dimming.

The World is pretty much the same as it was three years ago, but at least I have learned a lot about liquid crystal displays with dynamic light emission.

My colleagues and I started to work on this project in the beginning of 2010 pretty much from scratch, and I am quite impressed by what we have managed to do and learn during this time. What I feel now, besides relief for finally completing this thesis, is gratitude, and I wish to thank all the people that have helped me to get here.

I wish to thank my supervisor, Prof. Søren Forchhammer, for accepting my application and giving me the opportunity to pursue my Ph. D. at DTU. He has led me through the project with wisdom and competence, helping me to develop as a researcher and deepen my knowledge, and offered me very useful feedback on my work.

I wish to thank the Danish Strategic Research Council for funding the project and Bang & Olufsen, in the persons of Jesper Meldgaard Pedersen, Ben Verbraak and Søren Bech, for the precious support and contribution throughout the project. I also wish to thank the members of the advisory board of the project: Andrew Perkis, Richard Salmon, Huid de Ridder, Xiaolin Wu and Paul-Michael Petersen.

I wish to thank the co-authors of the papers I have worked on during the project; particularly, I wish to thank Jari Korhonen for guiding me at the beginning of my time at DTU and for later being my co-supervisor, Ehsan Nadernejad for being my “Ph. D. brother” and spending a lot of time together running experiments and revising drafts through most

of my Ph. D., and Claire Mantel for the outstanding contribution she brought since when she joined our group. I want to thank Ehsan and Claire one more time for being so kind to review and proofread this thesis.

I wish to thank my colleagues of the Coding and Visual Communication Group at DTU Fotonik for making my time there more enjoyable. A particular thanks to Marco Zamarin, Ann Ukhanova and Matteo Salmistraro for all the great time and talks together.

I wish to thank again Prof. Xiaolin Wu for hosting me in his group at McMaster University in Hamilton, ON (Canada) from August to December 2012. I have had a great and very productive time there, thanks to him and to my co-authors Xiao Shu and Liangbao Jiao. I am very thankful to Prof. Wu for offering me this opportunity.

I wish to thank the Otto Mønsted and Oticon funds for sponsoring my participation to international conferences and my external stay at McMaster University. Without their support, this might not have been possible.

I wish to thank Federica for the loving support and patience she has showed me in the last two years, especially during the last six months when she had to tolerate my absence or absent presence. That's the last Ph. D. I ever do, I promise!

I finally wish to thank all my family and all my friends, close and far, for accepting my distance but not letting connections go. A particularly grateful thanks to my mother and father, for which this is even harder, and are always there for me.

Thank you.

Nino Burini  
March 2013  
Kongens Lyngby  
Denmark

# List of Ph. D. Publications

This Ph. D. project has resulted in 11 peer-reviewed publications, including 3 journal papers and 8 conference contributions [1]–[11], listed below.

- [1] S. Forchhammer, M. Danieli, **N. Burini**, M. Zamarin, and A. Ukhanova, “Maximizing entropy of image models for 2-D constrained coding”, in *Proceedings of Workshop on Information Theoretic Methods in Science and Engineering (WITMSE)*, 2010.
- [2] J. Korhonen, **N. Burini**, S. Forchhammer, and J. M. Pedersen, “Modeling LCD displays with Local Backlight Dimming for Image Quality Assessment”, in *Proceedings of SPIE*, vol. 7866, 2011, p. 786 607.
- [3] **N. Burini**, E. Nadernejad, J. Korhonen, S. Forchhammer, and X. Wu, “Speedup of Optimization-based Approach to Local Backlight Dimming of HDR Displays”, in *Proceedings of SPIE*, vol. 8436, 2012, 84360B.
- [4] J. Korhonen, **N. Burini**, J. You, and E. Nadernejad, “How to Evaluate Objective Video Quality Metrics Reliably”, in *International Workshop on Quality of Multimedia Experience (QoMEX)*, Jul. 2012, pp. 57–62.
- [5] **N. Burini**, E. Nadernejad, J. Korhonen, S. Forchhammer, and X. Wu, “Image Dependent Energy-Constrained Local Backlight Dimming”, in *IEEE International Conference on Image Processing (ICIP)*, Sep. 2012, pp. 2797–2800.

- [6] E. Nadernejad, **N. Burini**, J. Korhonen, S. Forchhammer, and C. Mantel, “Adaptive Local Backlight Dimming Algorithm Based on Local Histogram and Image Characteristics”, in *Proceedings of SPIE*, vol. 8652, 2013, p. 86520V.
- [7] E. Nadernejad, J. Korhonen, S. Forchhammer, and **N. Burini**, “Enhancing Perceived Quality of Compressed Images and Video with Anisotropic Diffusion and Fuzzy Filtering”, *Signal Processing: Image Communication*, vol. 28, no. 3, pp. 222–240, Mar. 2013.
- [8] E. Nadernejad, **N. Burini**, and S. Forchhammer, “Adaptive Deblocking and Deringing of H.264-AVC Video Sequences”, in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2013, pp. 2508–2512.
- [9] **N. Burini**, X. Shu, L. Jiao, S. Forchhammer, and X. Wu, “Optimal Backlight Scanning for 3D Crosstalk Reduction in LCD TV”, in *IEEE International Conference on Multimedia & Expo (ICME)*, To be presented, 2013.
- [10] **N. Burini**, E. Nadernejad, J. Korhonen, S. Forchhammer, and X. Wu, “Modeling Power-constrained Optimal Backlight Dimming for Color Displays”, *IEEE/OSA Journal of Display Technology*, 2013. DOI: 10.1109/JDT.2013.2253544.
- [11] C. Mantel, **N. Burini**, E. Nadernejad, J. Korhonen, S. Forchhammer, and J. Meldgaard Pedersen, “Controlling power consumption for displays with backlight dimming”, *IEEE/OSA Journal of Display Technology*, 2013. DOI: 10.1109/JDT.2013.2260131.

This thesis is based on 5 of these publications [2], [3], [5], [9], [10], reported in Appendix A. Chapter 2 is based on [2], [3], [5], [10], while Chapter 3 is based on [9]. The other publications are briefly described below. Two papers on post-processing for artifact removal in image and video [7], [8] are reported in Appendix A as well.

## Maximizing Entropy of Image Models for 2-D Constrained Coding [1]

This paper considers entropy estimation and maximization in two dimensional fields for applications in 2-D constrained coding. The cases

of Markov random fields and of Pickard random fields are considered. Pickard random fields are bi-dimensional finite rectangular fields where each line and each row is a Markov chain. Pickard random fields are used to construct fields with spacial constraints, like the hard-square constraint and the no uniform  $2 \times 2$  square constraint. The entropy of these fields is estimated or bounded and compared to other estimations.

## **How to Evaluate Objective Video Quality Metrics Reliably [4]**

Image and video quality can be measured objectively using a large set of metrics, the most famous probably being Peak Signal to Noise Ratio (PSNR) and Structural Similarity (SSIM) [12]. However subjective quality is ultimately the most important, to the point that an objective metric is considered better than others if it can predict subjective evaluations more precisely. It is however not simple to determine the correlation between subjective and objective quality, because many factors can affect the reliability of the subjective data. By taking them into account it is possible to make more precise studies, or at least to know the degree of uncertainty of the conclusions that can be drawn from the analysis of such data.

## **Adaptive Local Backlight Dimming Algorithm Based on Local Histogram and Image Characteristics [6]**

This paper introduces a local backlight dimming algorithm using local histograms to determine the backlight level. The algorithm consists of three steps detailed. In the first step, segments are categorized in three groups depending on the average luminance. In the second step, the local histograms are built on the max RGB value of each pixel, then the initial intensity of each backlight segment is set according to a percentile of the histogram. Finally, the intensity of the segment is adjusted based on the classification done in the first step.

## **Enhancing Perceived Quality of Compressed Images and Video with Anisotropic Diffusion and Fuzzy Filtering [7]**

This paper presents two techniques to remove blocking and ringing artifacts through the combined use of anisotropic diffusion equations and fuzzy filtering. The first proposed method is designed for still images, the second for video sequences. Fuzzy filtering has proven effective in deblocking and deringing, while anisotropic diffusion is commonly used for image enhancement as it follows the directional nature of blocking and clipping artifacts. The filters have been designed to be adaptive to the image and video content.

The performance of the proposed approaches has been compared against other methods by using different objective quality metrics and a subjective comparison study. The results indicate that the proposed algorithms achieve better artifact reduction than other methods, on both still images and video sequences, also for H.264/AVC compressed video.

## **Adaptive Deblocking and Deringing of H.264-AVC Video Sequences [8]**

This paper presents a method to reduce blocking and ringing artifacts in H.264/AVC coded video. Deblocking is done with a decision mode-based algorithm using local block characteristics and a quality metric for I, P and B frames. Deringing is then obtained through an adaptive bilateral filter. Objective and subjective measurements show that the proposed algorithm effectively reduces artifacts and outperforms other methods.

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Motivation . . . . .	1
1.1.1	Dynamic Backlight Dimming in LCD . . . . .	2
1.1.2	Backlight Scanning for 3D Crosstalk Reduction . . . . .	2
1.2	Main Contributions . . . . .	3
1.3	Goals and Structure of the Thesis . . . . .	4
<b>2</b>	<b>LCD Backlight Dimming</b>	<b>5</b>
2.1	Light and Color . . . . .	5
2.2	LCD Technology . . . . .	10
2.2.1	The Backlight . . . . .	14
2.2.2	LEDs and Other Light Sources . . . . .	16
2.3	Backlight Dimming . . . . .	18
2.3.1	State of the Art . . . . .	19
2.3.2	Modeling . . . . .	23
2.3.3	Optimization Based Algorithms . . . . .	34
2.3.4	Gradient Descent . . . . .	47
2.4	Conclusion . . . . .	60
<b>3</b>	<b>LCD Backlight Scanning for 3D Crosstalk Reduction</b>	<b>63</b>
3.1	Modeling . . . . .	65
3.1.1	Timing of Scanning and Strobing . . . . .	65
3.1.2	Backlight . . . . .	66
3.1.3	Frames . . . . .	68
3.1.4	Pixel Transition and State . . . . .	68
3.1.5	Luminance . . . . .	69
3.2	Optimal Backlight Scanning . . . . .	71

---

3.2.1	Crosstalk . . . . .	72
3.2.2	Constraints . . . . .	72
3.2.3	The Optimization Problem . . . . .	73
3.3	Experiments and Results . . . . .	73
3.4	Conclusion . . . . .	75
<b>4</b>	<b>Conclusion and Future Work</b>	<b>79</b>
<b>Appendix A</b>	<b>Ph. D. Publications</b>	<b>83</b>
A.1	Modeling LCD Displays with Local Backlight Dimming for Image Quality Assessment . . . . .	85
A.2	Speedup of Optimization-based Approach to Local Back- light Dimming of HDR Displays . . . . .	97
A.3	Image Dependent Energy-constrained Local Backlight Dim- ming . . . . .	113
A.4	Enhancing Perceived Quality of Compressed Images and Video With Anisotropic and Fuzzy Filtering . . . . .	119
A.5	Adaptive deblocking and deringing of H.264/AVC video sequences . . . . .	141
A.6	Optimal backlight scanning for 3D crosstalk reduction in LCD TV . . . . .	149
A.7	Modeling power-constrained optimal backlight dimming for color displays . . . . .	157
<b>Appendix B</b>	<b>Testsets</b>	<b>169</b>

# List of Figures

2.1	Illustration of photometric concepts . . . . .	7
2.2	Wavelength sensitivity for rods and cones in the Human Visual System (HVS) (CC BY-SA 3.0 License by Maxim Razin, via Wikimedia Commons) . . . . .	8
2.3	Color matching functions of the CIE 1931 color space (CC BY-SA 3.0 License, via Wikimedia Commons) . . . . .	8
2.4	CIE 1931 chromaticity diagram (CC BY-SA 3.0 License, by Sakuranbo via Wikimedia Commons) . . . . .	10
2.5	Basic LCD structure and RGB sub-pixels (CC BY-SA 3.0 License, by Peter Halasz via Wikimedia Commons) . . . . .	11
2.6	Twisted Nematic LCD pixel (CC BY-SA 3.0 License, via Wikimedia Commons, courtesy of M. Schadt) . . . . .	12
2.7	Backlight classification . . . . .	16
2.8	Basic structure of LCD with local backlight . . . . .	24
2.9	Linear modeling of leakage . . . . .	26
2.10	Example of PSF based light diffusion . . . . .	27
2.11	Example of modeled and measured backlight . . . . .	28
2.12	Brightness compensation . . . . .	29
2.13	Perceptually uniform luminance . . . . .	32
2.14	Performance of perceptual weighting of the error . . . . .	39
2.15	$\ell_1$ optimization with power penalization . . . . .	40
2.16	$\ell_2$ color optimization on direct-lit screen . . . . .	41
2.17	Impact of downscaling . . . . .	42
2.18	MSE comparison for gradient descent . . . . .	50
2.19	LabPSNR comparison for gradient descent . . . . .	51
2.20	Leakage comparison . . . . .	52
2.21	Example of pixel error function . . . . .	55

2.22	MSE performance of block-based gradient descent . . . . .	57
2.23	MSE performance of block-based gradient descent (Zoom)	58
2.24	PSNR performance of block-based gradient descent . . . . .	59
2.25	PSNR performance of block-based gradient descent (Zoom)	60
3.1	Time chart of backlight scanning and strobing . . . . .	67
3.2	Pixel state function . . . . .	70
3.3	Average pixel state . . . . .	70
3.4	Backlight scanning experiment . . . . .	76
3.5	Backlight scanning control signals . . . . .	77
B.1	“ICIP” testset . . . . .	170
B.2	“ICIP” testset, grayscale . . . . .	171
B.3	Kodak True Color testset . . . . .	172
B.4	Kodak True Color testset, grayscale . . . . .	173
B.5	Four images for experiments . . . . .	174
B.6	Video testset . . . . .	175

# List of Tables

2.1	Performance physical optimization . . . . .	36
2.2	Physical optimization with power penalty . . . . .	38
2.3	Physical optimization, color . . . . .	40
2.4	Subset optimization . . . . .	45
2.5	Min-Max sub-pixel optimization . . . . .	47
2.6	MSE, leakage and clipping . . . . .	52
2.7	Comparison of backlight dimming algorithms . . . . .	53
2.8	Time performance of the block-based gradient descent . .	59



# Chapter 1

## Introduction

### 1.1 Motivation

The recent developments in information and telecommunication technology have favored the ubiquitous adoption of digital imaging and video. Anywhere, anytime, digital images and videos are being captured, coded, stored, transmitted, decoded and displayed. From capture to display, many factors affect the final visual quality. Compression of visual signals, particularly video, has become a common activity, often necessary to keep up to the growing requirements of visual content consumers, who demand higher resolution, lower bitrates and higher visual quality. Examples are digital Television (TV) broadcasting or video streaming from the Web using video codecs like MPEG-2 or H.264/AVC. While reducing the amount of bits necessary to store or transmit a visual signal, compression can introduce artifacts that negatively affect the perceived quality. The final visual quality is also significantly affected by the display technology. Several kind of displays exists, each with advantages and flaws, and they can have different sizes and different contexts of use, e.g. a mobile phone or a TV. The peculiar characteristics of every display technology can be exploited to yield a better rendition of the visual signals.

Nowadays, digital TV sets are very advanced and include systems and algorithms to improve the visual quality by processing the coded signals input to the system. In the case of Liquid Crystal Display (LCD), LED backlights can be varied adaptively to improve contrast and reduce

power consumption.

### 1.1.1 Dynamic Backlight Dimming in LCD

LCD is today the most common display technology in the TV sector. It has gradually eroded the market share of Plasma Display Panels (PDPs) and will remain the main player until other promising technologies (e.g. Organic Light Emitting Diode (OLED) displays) become more mature. LCDs are relatively cheap, bright and thin, but competing technologies (e.g. PDP) tend to render deeper blacks. The black level affects the display contrast ratio, an important aspect of display quality. The low contrast of LCD can be improved thanks to *backlight dimming*. The backlight is the part of the LCD that emits light and is the responsible of the poor black level; recent backlights commonly use Light Emitting Diodes (LEDs) as light sources. By adaptively dimming it to the image content, the black level can be reduced and the contrast ratio increased. Dimming the backlight also reduces the power consumption of the display, which is very important considering that modern screens are required to be more and more energy efficient by national regulations.

Local dynamic backlight dimming gives the opportunity to increase contrast and reduce power consumption, but it can introduce visual artifacts (i.e. clipping, haloes, flickering) that can negate the advantages of the technique if not taken care of. By modeling LCD systems with local backlight, it is possible to design advanced algorithms that find optimal tradeoffs between image quality, power consumption and complexity.

### 1.1.2 Backlight Scanning for 3D Crosstalk Reduction

Typically, LCDs are used to display Two Dimensional (2D) content, but it is possible to use them to deliver stereo vision of Three Dimensional (3D) content, for example using active shutter glasses. However, this setup can produce high levels of *crosstalk*, that is the imperfect isolation of the left and right view that generate the impression of stereo vision. This is caused by two aspects of a typical LCD: the backlight is conventionally always turned on and the pixels are refreshed one by one in sequence, causing previous and current frame to be partly displayed simultaneously. An intelligent use of the backlight, as in the case of local backlight dimming, can reduce the crosstalk level. This is done by *scan-*

*ning* the local backlight to follow the pixel refresh. However, backlight scanning reduces luminance, which is typically scarce when LCDs are used for stereo 3D vision. It is therefore very important to find a way to scan the backlight to reduce crosstalk but keeping the luminance at reasonable levels.

## 1.2 Main Contributions

This Ph. D. thesis focuses on the visual quality of digital video signals displayed on TV. In particular, the main interest is directed to LCD with local dynamic backlight. The goal of the thesis is to show that visual quality is also affected by the display; that the use of backlight dimming can effectively improve image quality and energy efficiency; and that backlight scanning can be used to reduce crosstalk in stereo 3D LCD with active shutter glasses.

The main focus of the Ph. D. project has been on local backlight dimming. An initial model of backlight dimming systems and an optimization-based backlight dimming algorithm built on it [13] have been extended by introducing the possibility to penalize power consumption, to reduce perceptual errors [5], and to optimize color images [10]. We have shown that it is possible to obtain optimal results using only a subset of the image pixels [3]. We have also developed an alternative approach, based on Gradient Descent (GD) search, which takes an initial backlight as input and improves it iteratively given a cost function. This algorithm allows to directly optimize the perceptual result and determine the backlight giving the best visual result. It is versatile, since the number of iterations can be adapted to the context of utilization. It is possible to reduce the complexity of the approach by assuming block-wise uniform backlight in the screen; nearly optimal results can be achieved with a greatly reduced execution time.

Modeling backlight dimming made it simple to model backlight scanning for stereo 3D crosstalk minimization, which is partly based on similar principles [9]. Temporal variation of the backlight and the pixels was added to the model, to characterize the scanning process of the Liquid Crystal (LC) pixels, which is the main cause for crosstalk in the display. We have then formulated an optimization problem to minimize crosstalk at a given luminance level. The final results indicate that this can be

achieved by properly offsetting the signals controlling the scanning of the backlight.

### 1.3 Goals and Structure of the Thesis

This Ph. D. project has resulted in 10 peer-reviewed publications, including 2 journal publications and 8 conference contributions [1]–[10]. The publications are in the fields of display technology, mathematical modeling, image processing, video coding, visual quality assessment and information theory. Of these publications, 5 are part of this thesis [2], [3], [5], [9], [10] and are reported in Appendix A. Two papers on post-processing for artifact removal in image and video [7], [8] are reported in Appendix A as well.

Chapter 2 presents local backlight dimming. After a brief introduction to photo/colorimetry and LCD, the benefits and challenges of the technique are discussed, followed by the description of State of the Art algorithms. A model of LCD with local backlight dimming, including important aspects like light diffusion and perception, is introduced and used as foundation to design optimization-based algorithms. Many such algorithms are presented, starting from a simpler version and gradually extending to improve the final result or allow new solutions like adjustable power consumption. To conclude, an algorithm based on GD search and another one using local histograms are described.

Chapter 3 deals with backlight scanning for crosstalk reduction when stereo 3D content is shown using an LCD and active shutter glasses. A mathematical model of the system is introduced, followed by the formulation of an optimization problem that minimized crosstalk at a given luminance level.

Finally, Chapter 4 draws the conclusion of the thesis work.

## Chapter 2

# LCD Backlight Dimming

This Chapter is dedicated to backlight dimming techniques, which allow to reduce the power consumption of Liquid Crystal Displays (LCDs) and to increase their contrast at the same time. The Chapter is based on [2], [3], [5], [10].

After a brief overview photometry and colorimetry principles and an introduction to LCD technology, backlight dimming will be discussed. The discussion will include the description of algorithms which constitute the State of the Art, followed by the proposal of a model of backlight dimming enabled LCDs. This model serves as framework to design backlight dimming algorithms based on optimization techniques.

### 2.1 Light and Color

Light is an electromagnetic radiation that can produce a visual sensation. This section will consider light which can be perceived by the Human Visual System (HVS), also called "visible light". The wavelengths of visible light range approximately from 400 nm to 700 nm. Other definitions include ultra-violet (below 400 nm) and infra-red (above 700 nm) light.

There are several units of measure related to light. They can be classified in *radiometric units*, which consider radiation in one part of the spectrum, and *photometric units*, which consider visible radiation relatively to the sensitivity of the HVS. The major photometric units of measure are:

**Luminous flux** visible power or light energy per unit of time; it is measured in lumens (lm).

**Luminous intensity** luminous flux per solid angle emitted or reflected from a point; it is measured in candelas ( $\text{cd} = \frac{\text{lm}}{\text{sr}}$ ).

**Luminance** luminous intensity per unit of area projected in a given direction; it is measured in candelas per square meter or nits ( $\text{nit} = \frac{\text{cd}}{\text{m}^2}$ ). It can be seen as the amount of light emitted by a surface and falling in a specific solid angle. It is often used to measure the light emitted by flat surfaces, like displays. For example, an LCD TV can have a peak luminance of  $500 \frac{\text{cd}}{\text{m}^2}$ .

**Illuminance** luminous flux per unit area incident on a defined surface; it is measured in lux ( $\text{lux} = \frac{\text{lm}}{\text{m}^2}$ ).

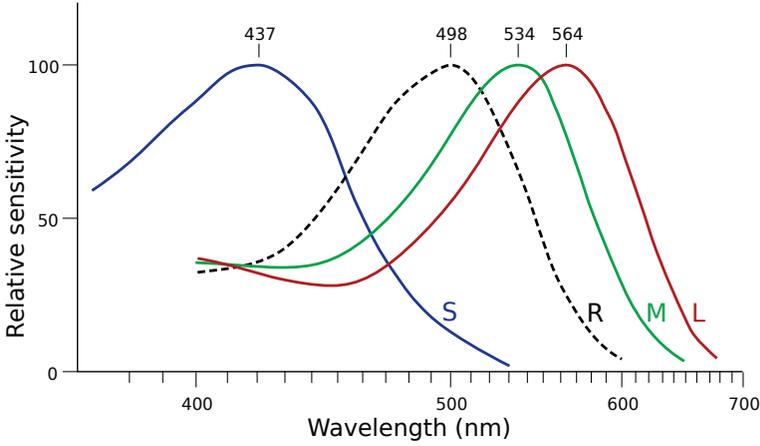
The concepts of these measurements are illustrated in Figure 2.1. The term **brightness**, often misused [14], refers to the subjective perception of luminance by humans. Observers can perceive that objects are more or less bright, but brightness cannot be measured.

As mentioned, the HVS is sensible to electromagnetic waves with wavelengths ranging approximately from 400 nm to 700 nm. These waves generate the sensation of color. The retina has four kind of sensors: one type of "rods" and three types of "cones". Rods sense in presence of less then 100 nits and their peak sensitivity is at 498 nm. Cones sense color when luminance is above 0.001 nits; their peak sensitivities are about 437 nm (S or "blue" cones), 533 nm (M or "green" cones) and 564 nm (L or "red" cones). Figure 2.2 shows the sensitivity of all sensors as a function of wavelength.

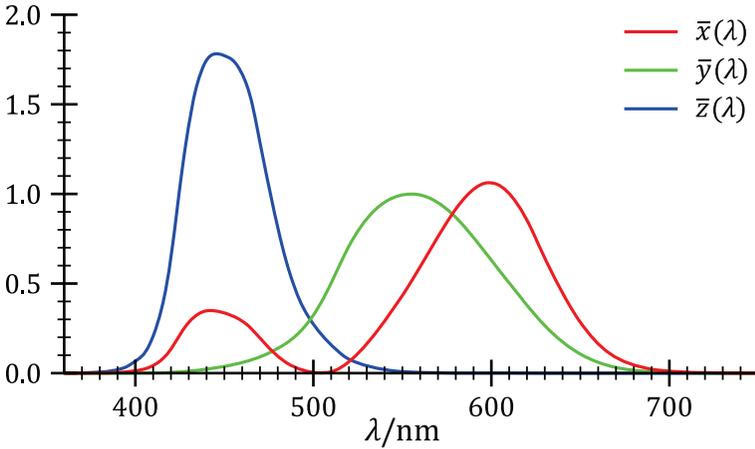
When only rods are sensing, the vision regime is called *scotopic* and only shades of gray are perceived. When rods and cones are both active the regime is *mesopic* and there is limited color perception. When only the cones are active the regime is *photopic*.

Colorimetry is the science of describing in physical terms the human perception of color. One of the most important color spaces is the *CIE 1931 color space*, which is based on tristimulus values derived from the sensitivity to colors of the HVS. The tristimulus values are called X, Y and Z and are functions of the *spectral distribution* of the colored light and of the *color matching functions*. These functions map the response





**Figure 2.2:** Wavelength sensitivity for rods (dashed line) and cones in the HVS.



**Figure 2.3:** Color matching functions of the CIE 1931 color space.

It is possible to distinguish, in a color, luminance and chromaticity. Luminance is represented by  $Y$ , while chromaticity describes the hue and saturation of the color. Chromaticity is described by the derived values  $x$  and  $y$  and can be plotted in a diagram:

$$x = \frac{X}{X + Y + Z} \quad (2.4)$$

$$y = \frac{Z}{X + Y + Z} \quad (2.5)$$

$$z = 1 - x - y \quad (2.6)$$

The diagram is called the *CIE 1931 chromaticity diagram* (Figure 2.4) and contains all the colors averagely perceivable by a the HVS (or *gamut* of human vision). The curved border is called *monochromatic locus* and corresponds to pure (fully saturated) colors, which are composed of just one wavelength. The point (x=0.33, y=0.33) is called the *equal energy point*, which is taken as a reference to define the *purity* of a color C. Purity is determined by drawing a segment between from the equal energy point to the border crossing the color point and by calculating the ratio of the distances. The color on the border is also called *dominant color* of C.

When dealing with displays, one of the most significant metrics is the contrast ratio, also simply called contrast. Contrast is a measure of the difference between two luminance levels. The most common definitions of contrast are Weber's:

$$C = \frac{\Delta L}{L}, \quad (2.7)$$

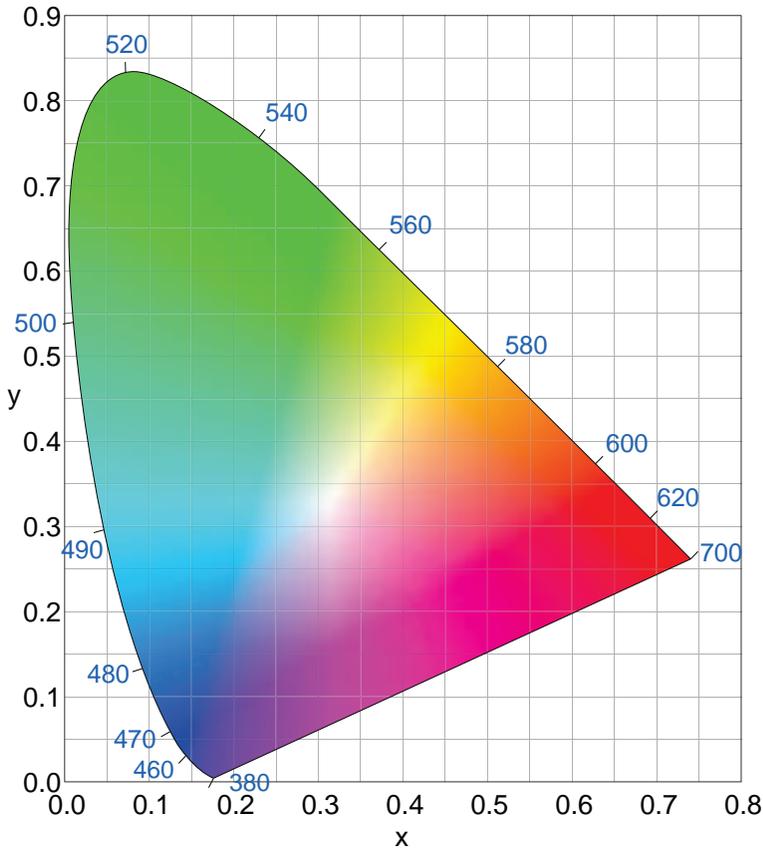
where  $L$  is the absolute luminance and  $\Delta L$  is the luminance variation, and Michelson's:

$$C = \frac{L_{max} - L_{min}}{L_{max} + L_{min}}, \quad (2.8)$$

where  $L_{max}$  and  $L_{min}$  are the maximum and minimum luminance, respectively. When concerning display, contrast is often measured as the ratio between the luminance of the brightest and the darkest emissions of the screen. In general terms, display contrast depends on what image is displayed and is the ratio between the brightest and the darkest pixel. The highest contrast can therefore be obtained when measured over black and white pixels. In some kind of displays, like some LCDs, the luminance of white and black pixels is not fixed. This leads to two definitions of contrast:

**Dynamic contrast** is the contrast between the highest and the lowest pixel brightness in a range of frames.

**Static contrast** is the contrast between the highest and the lowest pixel brightness in a single frame.

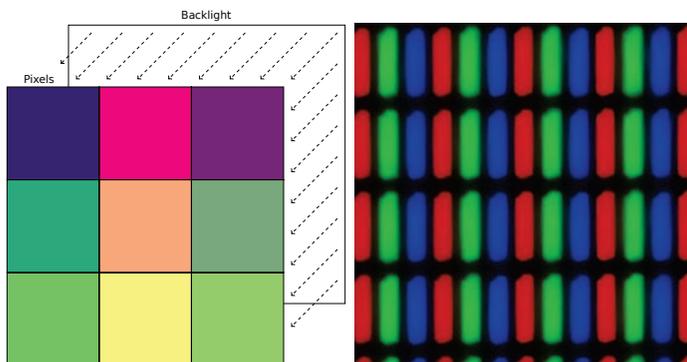


**Figure 2.4:** CIE 1931 chromaticity diagram. The numbers on the curve edge indicate the frequencies of pure colors.

In some vendor specifications, dynamic contrast can easily be infinite since it is sufficient to turn off the backlight to achieve such result (see the following Section). Static contrast is a more meaningful parameter and has a higher impact on image quality. In this thesis, the term contrast refers to static contrast, unless otherwise noted.

## 2.2 LCD Technology

This section presents the main characteristics and basic functioning of LCD technology, with particular focus on the backlight and on Light

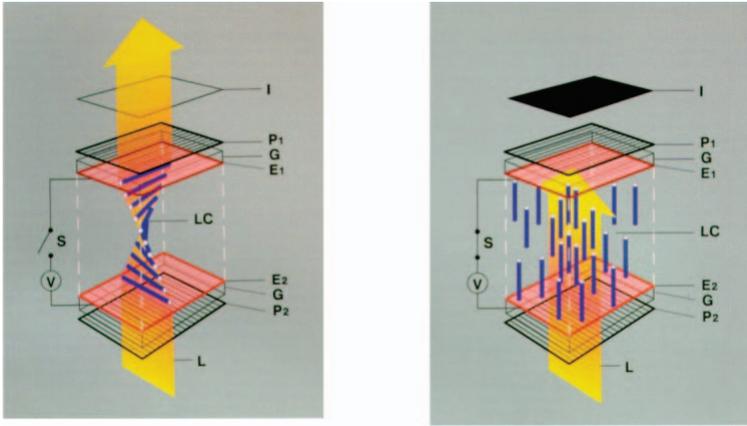


**Figure 2.5:** Basic structure of a LCD where the LCs are modulating and coloring the backlight (left) and detail of RGB LC sub-pixels (right).

Emitting Diodes (LEDs) as light source. LCDs are one of the most widespread display technologies, used on a large group of devices, from digital watches to mobile phones and Television (TV) screens. In this thesis, we will mainly consider the case of LCD TV.

In LCDs, the pixels are based on Liquid Crystals (LCs). The LC grid is placed in front of the backlight, which is the light source of the display. The LCs do not emit light themselves, but instead act as light filters with adjustable transmittance (the fraction of light which is let through), modulating the intensity of the backlight for each pixel. Each LC pixel is usually composed of three sub-pixels, each one with a primary color filter. The primary colors are usually Red, Green and Blue (RGB), but screens with four or more primary colors exist as well [15]. The LCs and the color filters allow each pixel of the LCD to render a wide range of colors and luminance values. Figure 2.5 includes a representation of sub-pixels.

The transmittance of the pixels of the LCD can be controlled with an electric signal. The LCs are placed between light polarizers oriented perpendicular to each other. If properly polarized, the light generated by the backlight goes through the first polarizer. When no voltage is applied on them, the LCs assume an helix-shaped structure, which “twists” the polarization of the light and lets it through the second polarizers. The result is a bright pixel. If instead a voltage is applied, the helix-shaped structure starts to lose its form and the light is twisted to a lesser extent.



**Figure 2.6:** Example of TN pixel in two states: OFF/white (left) and ON/black (right).

This results in less light going through the second polarizer and in a dimmer pixel. At a sufficiently high voltage, no twisting happens and no light goes through the second polarized, resulting in a dark pixel [16]. What has just been described is the type of LCDs based on Twisted Nematic (TN). Alternative technologies have been invented, like In-Plane Switching (IPS) and others.

However, the LCs cannot block all the light when this is intended. This goes under the name of *light leakage*. For simplicity, in this thesis light leakage is just called *leakage*.

The signal controlling the transmittance of LCD pixels do not have a linear behavior. In order to make a more efficient use of their limited bit depth, the signals are typically coded with a power function with  $\gamma$  at the exponent. The function is called Gamma and a common value of  $\gamma$  is 2.2, although others can be used [17].

Other popular display technologies exist. Before LCDs had taken over, the most popular kind of consumer TV displays were Cathode Ray Tubes (CRTs). Compared to them, LCDs are thinner, lighter, brighter and consume less power. However, CRTs have lower black level (the black level is the brightness of the black color when displayed, and it should be as low as possible). Today, the main competitor for the TV market are Plasma Display Panels (PDPs). Compared to LCD they have better black level and a larger potential screen size. LCDs are

however cheaper, brighter and consume less power. Over the years, manufacturers have managed to make LCDs with large diagonals less and less expensive, gradually eroding the PDPs market share. As for emerging technologies, Organic Light Emitting Diodes (OLEDs) are expected to take over the TV market some time in the future. OLEDs overcome many of the weak points of LCD, but are currently too expensive and with some limitations that make their use for large displays like TVs still inconvenient. They are however already present in the mobile screen sector.

The LCD technology is very attractive and has been very successful. There are however some flaws that make it inferior to competing technologies in some aspects. One of these is the poor black level, that is the excessively high brightness of the color black rendered on the display. This is caused by the LC leakage, which lets some unintended light through because of imperfections in the material and the design of the LCD pixels. Therefore, dark pixels look brighter than they should, causing a high black level. LCDs are also affected by motion blur, which is caused by the hold-type nature of the displays [18]. One more problem is that the time it takes to change the transmittance of a LC is generally not negligible. This, together with the fact that the pixels in a LCD are usually updated one-by-one sequentially and not simultaneously, increases crosstalk when LCDs are used to display stereo Three Dimensional (3D) content (see Chapter 3).

The LCD module in a TV was specified to consume about 150 watts of power in 2006 and about 70 in 2010 for a diagonal of 40 inch; the main LCD component responsible for power consumption is the backlight, which can account for up to 90% of the total energy usage [19]. LCD technology suffers from an intrinsic energy waste problem, since the light emitted by the light sources is dissipated by several components (light guides, diffusers, polarizers, filter, etc.) before it reaches the viewer. The waste is maximum for black pixels: the light is generated, but then it is blocked by the liquid crystals (except for a fraction of it, which leaks through undesirably). In order for the viewer to perceive a sufficient brightness, it is necessary for the light sources to be very powerful. One consequence of this is a significant generation of heat, which might affect the performance of the LCD and deteriorate image quality in the long term. Reducing power consumption is therefore beneficial in

many ways to LCDs. One more reason to save power is the recent enforcement of regulations limiting the allowed power consumption for TV sets in several countries [20]. Labeling systems favoring energy-efficient products have also been introduced, pushing TV manufacturers to make their TV sets less power hungry.

One solution to counter balance the flaws of LCDs is backlight dimming. The main principle of backlight dimming is that not all images require the same level of brightness, depending on their content, and that therefore it is possible to adaptively dim the backlight, when appropriate. This helps lowering the LCD black level and consequently the contrast ratio. Backlight dimming can also mitigate motion blur and reduce 3D crosstalk. Clearly, dimming the backlight will reduce its power consumption and consequently heating. There are of course challenges connected to this technique, like visual artifacts generated by changing the backlight intensity. Backlight dimming is extensively discussed in Section 2.3.

### 2.2.1 The Backlight

The backlight is the LCD component that provides light to the LCs. The light is generated by one or more light sources, which can be of several kinds (see Section 2.2.2); this work mainly considers the case of LED backlight. The light sources can be placed directly behind the LC pixels or at one or more of the sides. This allows to distinguish *direct-lit* and *edge-lit* backlights, respectively. Edge-lit backlight need a *light guide* to spread the light coming from the sides on the whole screen. This is not the case for direct-lit backlights. The light distribution is then made more uniform across the screen by using light *diffusers*. After this, the light reaches the LC pixels and finally the viewer.

Besides the classification by light-source and by direct-lit/edge-lit, backlights can be divided into *local* and *global*. Local backlights are divided in *segments*, each one including one or more light sources and each one independently controllable. In global backlights, a change in light intensity concerns the whole screen. This can happen with one single light source or with many light sources controlled by a single signal. They can be modeled as having a single segment as large as the display. Conventional Cold Cathode Fluorescent Lamp (CCFL) based

backlights are usually global and edge-lit, with the lamp placed at the lower side. More recent high-end LED based backlights are direct-lit and local. Manufacturers are also producing edge-lit LED backlight where the LEDs are placed on the left and right side, to contain costs by using less light sources.

Local backlights allow to vary the light intensity spatially, which enables local backlight dimming (see Section 2.3). Depending on the layout of the segments, local backlight can be classified by the number of dimensions. When the segments span across one of the dimensions of the screen, the backlight is One Dimensional (1D) (either vertical or horizontal). When the segments are placed in a more grid-like structure (usually square or hexagonal), the backlight is classified as Two Dimensional (2D). Some 2D backlights can also change the local light color, thanks to RGB light sources; these are called RGB 2D backlights or 3D backlights (color is considered a dimension). Figure 2.7 shows examples of different kinds of backlights. The following list summarizes the classification between global and local backlights.

**Global backlight:** a change in light intensity regards the whole screen; in other words, there is only one backlight segment as big as the screen.

**Local backlight:** the backlight is divided in separate segments that can be dimmed independently; local backlight can be further classified in

**1D backlight:** has horizontal or vertical segments that span across the whole screen.

**2D backlight:** backlight segments are arranged in a grid, usually square or hexagonal.

**RGB 2D backlight:** also known as 3D backlight, is an extension of 2D backlights where regions are not illuminated by a single white LED but by three RGB LEDs that spread out colored light.

When using backlight dimming, more dimensions usually bring better contrast and save more energy. The number of segments is also influential, as in general the more the better. However, the price for this is a higher production and driving cost [21].



**Figure 2.7:** Examples of backlights depending on the type and position of backlight segments. From top to bottom, left to right: input image, global (0D) backlight, horizontal (1D) backlight, vertical (1D) backlight, 2D backlight, 3D (2D+color) backlight. The two bottom displays are examples of the backlight generated from the input image in case the backlight is 2D direct-lit (left) or edge-lit (right).

## 2.2.2 LEDs and Other Light Sources

For a long time, the most convenient light source for LCD backlight were CCFLs, which are built with glass tubes containing two electrodes

at the ends and filled with mercury (Hg) vapor. The internal side of the glass tube is covered with fluorescent substances. When a voltage is applied to the electrodes, electrons in the tube are accelerated and ionize the Hg atoms; this leads to more atoms being hit by electrons and by energy radiating from the impacts. This energy activates the fluorescent materials in the glass tube, consequently emitting visible radiation. Other kinds of light sources have been used for backlights, in particular External Electrode Fluorescent Lamps (EEFLs), Flat Fluorescent Lamps (FFLs), Hot Cathode Fluorescent Lamps (HCFLs) and Field Emission Lamps (FELs) [19]. In the recent years, LEDs have been used more and more as light source for backlights, replacing the competing technologies which have been listed, particularly CCFLs [22].

LEDs are semiconductor devices that can emit light and have several advantages. LEDs response time is less than a microsecond, while that of CCFLs is in the order of milliseconds [22]. This short response time enables techniques which would otherwise be impossible, like backlight scanning and strobing/blinking, used to mitigate motion blur in LCDs [18] or to reduce crosstalk in field-sequential 3D LCD TV (see Chapter 3). The lifespan of a LED exceeds 50000 hours in average, that is 4 or 5 times more than CCFLs [22]. CCFLs contain mercury, which is intoxicating and whose usage has been restricted in several countries; LEDs do not contain mercury. Being semiconductors, LEDs driving circuits are easier to design and generally simpler. Moreover, they are more shock resistant and generally more versatile. For instance, it is simpler to design local backlights using LEDs. Additionally, in the case of RGB LEDs, the spectrum of the backlight is sharper compared to that of CCFLs, resulting in purer colors and larger gamut [22].

Thanks to recent research efforts, LEDs have become one of the most efficient lighting technology. Furthermore, research on LEDs is still in a relatively early stage compared to the mature CCFL technology, so it is expected that their advantages will improve even further and their price will decrease. This thesis focuses on LED backlight, therefore we assume the light sources to be LEDs.

## 2.3 Backlight Dimming

This section discusses backlight dimming, a technique that allows to reduce power consumption and improve image quality on LCDs. As a reminder to what was mentioned in Section 2.2.1, the main idea is to match the intensity of the backlight to the image that is displayed. The reason for this is that not all images require the same amount of light, since some are darker than others. With local backlights, it is even possible to vary the light intensity spatially according to the image content, e.g. if the image contains both dark and bright areas, the backlight can be dimmed only in the dark ones. Local backlights often use LEDs as light sources; for this reason, in this thesis the term LED is often used to refer to a backlight segment. The reduced backlight emission can be compensated by increasing the LC pixel transmittance accordingly, in order to obtain the same output luminance.

One clear benefit from backlight dimming is reduced power consumption, as the backlight is not kept at full power all the time. This is very important for the energy efficiency of LCD TV, because normally the backlight is the component requiring the most power. The reason for this is that, before reaching the viewer, up to 90% of the light is dissipated by several components, like light guides, diffusers, polarizers, LCs, filters, etc. [19]. The light sources must therefore emit very intense light, consequently consuming a large amount of power. Energy efficiency is very important for TV manufacturers because new regulations tightening the allowed power consumption for TV screens and monitors have been enforced in several parts of the world, including the EU [20]. An additional benefit of lower light emission is the reduction of heat generation, which decreases mechanical distortions of the screen caused by high temperatures [23].

Backlight dimming also helps to reduce a typical problem of LCD: light *leakage*. Leakage is caused by imperfections in the LCs preventing a complete obstruction of light when this is required, i.e. black pixels, and is particularly visible from wide viewing angles. This raises the black level of the screen, making dark pixels brighter than desired. This is the reason for the relatively poor contrast ratio of LCD, compared to competing display technologies. When the backlight is dimmed, the amount of light leaking through dark pixels is reduced and consequently

the contrast ratio is increased. Varying the intensity of the backlight also allows to raise the number of distinct luminance levels that the display can emit, and therefore enable High Dynamic Range (HDR) rendering.

While backlight dimming can deliver undeniable benefits, it has some problems. If the backlight is dimmed to an excessive degree, some pixels might not be able to increase their transmittance enough to compensate for the reduced emission of light, causing them to appear darker than intended. This creates an artifact called *clipping*; the pixels that suffer from it are called clipped. If the backlight intensity changes too frantically, for example when displaying a sequence of frames part of a video, the user might experience the annoying artifact of *flickering* over time. In local backlights, the light generated by several LEDs typically mixes in the light guides, if any, and the diffuser. The results is a many-to-many relationship between LEDs and pixels, where each LED contributes to the luminance of many pixels and where each pixel receives light from many LEDs. Because of this, bright and dark pixels are often in conflict, particularly when in the same region: it might not be possible to obtain a backlight intensity which is optimal for both, because the high luminance required by the former would cause leakage in the latter; conversely the low luminance that would avoid leakage in the latter would clip the former. In particular, the presence of a group or cluster of bright pixels surrounded by dark pixels would induce a *halo* artifact, caused by leakage being more evident in the dark pixels close to the bright cluster.

Several algorithms have been proposed to calculate dynamic backlights. All of them need to find trade-offs between image quality, power saving and complexity. Some favor one aspect, some another. Many State of the Art algorithms are described in Section 2.3.1. These include proposals that have been developed during the Ph. D. project which is the object of this thesis; they are presented in Sections 2.3.3 and 2.3.4. The model of backlight dimming enabled LCDs used to design these algorithms is presented in Section 2.3.2

### 2.3.1 State of the Art

This section presents a selection of backlight dimming algorithms that define the State of the Art. These algorithms have been used for compar-

ison against the algorithms developed during the Ph. D. project object of this thesis. For simplicity, after having been presented, the algorithms are referred to with the name of the first author of the corresponding publication.

The simplest backlight dimming algorithms use simple global or local image characteristics, for example the maximum or average pixel value, to determine the backlight. The Max algorithm sets the intensity of each LED to the maximum pixel value of the corresponding segment, while the Average (Avg) algorithm uses the mean value [24]. The Square-root (Sqrt) algorithm uses the square root of the normalized average pixel value, i.e. valued between 0 and 1 [25]. These algorithms are the most simple, but are not reliable because the Max might not bring any energy saving, especially for large segments, and is very sensitive to noise and prone to flickering, while the Avg tends to produce excessively dim backlights. The Sqrt is a trade-off between the two, but still too basic.

Other algorithms, like the one by Cho et al. [26] and the one by Zhang et al. [27] attempt to use more advanced image statistics to obtain better results. Both algorithms adopt the strategy of adding a correction term to the average luminance required by each segment:

$$r_k = g_{avg,k} + corr, \quad (2.9)$$

where  $r_k$  is the intensity of LED  $k$  and  $g_{avg,k}$  is the average luma value of the pixels of the target image ( $y$ ) in segment  $k$ . The two algorithms differ in the calculation of the correction term  $corr$ : in Cho [26] it is calculated as

$$corr = \frac{1}{2} \times \left( d + \frac{d^2}{2^n} \right), \quad (2.10)$$

where  $d = (g_{max,k} - g_{avg,k})$ ,  $g_{max,k}$  is the maximum luma value of the pixels in segment  $k$  and  $n$  is the bit depth of  $y$ ; in Zhang [27]  $corr$  is calculated as

$$corr = \begin{cases} 0 & \text{if } \sigma_k = 0 \\ \left( 1 - \frac{\sigma_{avg}^2}{\sigma_k^2} \right) \times d & \text{if } \sigma_k \neq 0 \end{cases}, \quad (2.11)$$

where  $\sigma_k^2$  is the luminance variance of segment  $k$  and  $\sigma_{avg}^2$  is the variance when the maximum backlight luminance is equal to the average luminance of segment  $k$ . The Cho algorithm corrects the backlight by

considering the local difference between maximum and average luminance, while the Zhang algorithm tries to compensate the loss of detail that an excessively low luminance would cause. The values of *corr* can be stored in a Look-Up Table (LUT) to make the algorithms faster.

Another algorithm, introduced by Nam [28], uses a combination of global and local image characteristics. Depending on the local and global maximum and average luma values, the LEDs are set as follows:

$$r_k = \begin{cases} \left(\frac{g_{m,k}}{2^n - 1}\right)^\gamma & \text{if } g_{m,k} \geq g_m \\ \left(\frac{g_{max,k}}{2^n - 1}\right)^\gamma & \text{if } (g_{m,k} < g_m) \wedge (g_{max,k} \leq g_{avg}) \\ \left(\frac{g_{avg}}{b}\right)^\gamma & \text{if } (g_{m,k} < g_m) \wedge (g_{max,k} > g_{avg}) \end{cases}, \quad (2.12)$$

where  $g_{m,k} = \frac{g_{max,k} + g_{avg,k}}{2}$ ,  $\gamma$  is the gamma of the display,  $g_{max}$  is the maximum luma value of image,  $g_{avg}$  is its average luma value,  $g_m = \frac{g_{max} + g_{avg}}{2}$  and

$$b = (2^n - 1) \times \left(1 - \frac{1 - \frac{g_{avg}}{g_m}}{g_{max} - g_{avg}}(g_{max,k} - g_{avg})\right). \quad (2.13)$$

The author also describes an algorithm-specific way to compensate the LCs for the light loss due to dimming [28].

Kim et al. [29] proposed an iterative local backlight dimming algorithm considering the characteristics of neighboring image segments to set the intensity of each LED. The authors define two measures of the severity of leakage and clipping artifacts in each block. The algorithm iteratively calculates the LED duty cycle that achieves a certain ratio between the two artifacts. This ratio is defined by a segment-specific weight, which is calculated using the maximum pixel value of the segment and the average pixel value of the neighboring segments.

Several algorithms set the backlight using global or local histograms. Kang presented two global dimming algorithms that use the global histogram in one case and multiple local histograms in the other [30], [31]. Both methods take a target Peak Signal to Noise Ratio (PSNR) as input and return a result with equal or higher value. In [30], the global backlight is gradually dimmed (starting from full power) as long as the PSNR is above the target. The method in [31] is very similar, but in this case the image is divided in a certain number of blocks; the backlight

is gradually dimmed as long as the PSNR of each image block is above the target. Consequently, at the end each block has a PSNR higher or equal to the target, and so does the whole image.

In [32], Chen et al. initially set the LED intensities using local weighted histograms. The intensities are increased if the image is averagely dark but contains some bright pixels, to avoid clipping. Flickering is reduced by applying an adaptive Infinite Impulse Response (IIR) temporal filter. The adaptive term depends on the average gray value of the image and attempts to match the speed at which the scene changes. All the LED are then dimmed in a final global step to reduce light leakage.

Lai et al. [33] build local histograms from the maximum RGB value of each segment and use the gray level below which 75% of the pixels of the segment are to set the LED. The algorithm presented by Cho et al. in [34] calculates separate RGB histograms for each backlight segment. These histograms are used to reduce the intensity of each LED as long as the overall and maximum pixel luminance remain above a threshold value. Another histogram-based algorithm was presented by Lin et al. in [35]. This algorithm considers the global histogram as a Probability Density Function (PDF) and calculates a Cumulative Distribution Function (CDF) from it. The CDF is then inverted and used to map a weighted mean of the maximum and average pixel value of each backlight segment to the resulting LED intensity.

Other algorithms use knowledge of the light diffusion in the backlight to improve the precision of the calculations, typically at the cost of higher complexity [13], [36], [37]. The information of light diffusion is represented by the Point Spreading Function (PSF), also called light spreading function [35], which is how the light spreads on the diffuser plate from a light source. Each backlight segment has a characteristic PSF. The algorithm by Albrecht et al. [36] is a three-step iterative method that produces clipper-free results while reducing the power consumption of the backlight. During the first step, each LED is set to a lower bound determined by the image data and by the PSF of the LED. The second step is optional and iterative: for each iteration, the pixel requiring the most additional luminance to reach its target is found and the intensity of the most influential LED not already at full power is increased to reduce the gap. Iterations can continue for a defined number of steps or until all pixels receive enough light. In the third step,

each segment is scanned to find clipped pixels and the luminance of the corresponding LED is increased if any is found.

The algorithm presented by Hong et al. [37] determines, for each backlight segment, the uniform luminance level which would keep a clipping measure below a certain threshold. Then it finds the combination of LED intensities that keeps power consumption to a minimum, while keeping clipping below the threshold in all segments. To do this, the algorithm uses a block-based coarse PSF, which assumes uniform backlight inside one segment. This reduces accuracy as well as complexity.

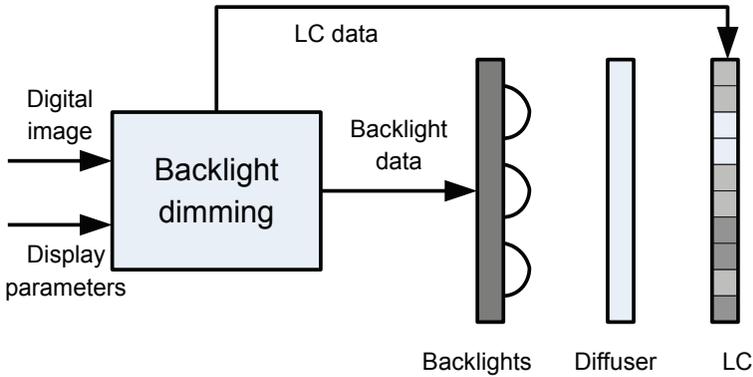
Somehow similarly to [31] and [37], Cho et al. [38] use histograms to determine the clipping error in each segment and set the LED luminance to match a target PSNR in each block. Because of the light diffusion from other LEDs, the luminance in each block is actually higher than required. To reduce power consumption further, the LED intensity of each segment is reduced if the amount of light coming from the neighbors is high enough.

Shu et al. [13] presented an optimization-based dimming algorithm that finds the best trade-off between clipping and leakage. This approach is at the foundation of those presented in Section 2.3.3 and will be discussed later.

### 2.3.2 Modeling

This section presents a model for LCDs capable of local backlight dimming that has been used to develop the algorithms presented in Section 2.3.3. Initial elements were first presented in [2] and [13], then extended and formalized in [5], [3] and [10]. The modeling needs to include important factors like the transmittance of the LC pixels, the backlight intensity for all pixels, light diffusion by PSF, leakage and human perception of luminance. Although LCDs with more primary colors exists, our model considers the common case of RGB LCD.

Figure 2.8 shows the simplified structure of an LCD with local backlight dimming. Generally, backlight dimming algorithms take the digital image to be displayed as input, together with other input parameters specifying the characteristics of the display (as light diffusion or leakage), and returns LED values and new pixel transmittances as output. The model assumes the input images to be in the sRGB color format [39].



**Figure 2.8:** Basic structure of a simplified LCD with local backlight. A backlight dimming algorithm takes a digital image and display parameters (like light diffusion or leakage) as input to deliver LED values and compensated pixel transmittances.

Some algorithms use the RGB values directly, while others first linearize them by applying a power function to the normalized RGB values, with an exponent of 2.2 which coincides almost perfectly with the gamma of the color space.

Backlight dimming algorithms define some sort of target luminance for each pixel or for a group of pixels; the groups usually coincide with the backlight segments. Some algorithms set the target luminance for each RGB sub-pixel, while others use only one value for the pixel, often based on its luma. Some algorithms calculate the target luminance from the linearized RGB values, others use the values directly.

### Luminance, Transmittance and Leakage

The physical luminance  $l$  of one pixel in a LCD with local backlight can be expressed as

$$l = bt, \quad (2.14)$$

where  $b$  is the intensity of the backlight behind the pixel and  $t$  is the transmittance of the pixel. Both  $b$  and  $t$  are normalized between 0 and 1. When  $b = 0$  there is no light behind the pixel, while  $b = 1$  means that it is at maximum intensity. In traditional LCDs without backlight dimming,  $b$  is set to 1 and the pixel luminance is determined by  $t$  only. Similarly to  $b$ ,  $t = 0$  means that the pixel transmittance is null and,

ideally, no light is let through, while when  $t = 1$  no light is blocked. However, because of the typical LCD leakage problem,  $t$  cannot be 0 in practice [13]. Leakage is modeled introducing the *leakage factor*  $\varepsilon$ , defined as the ratio of light leaking through the LCs when  $t = 0$  and  $b = 1$ ;  $\varepsilon$  is therefore the maximum physical leakage that can occur in a pixel. Considering leakage, the output pixel luminance equation acquires a second term proportional to  $\varepsilon$ ,  $b$  and  $(1 - t)$ :

$$l = bt + \varepsilon b(1 - t). \quad (2.15)$$

Leakage is proportional to  $(1 - t)$  because it is maximum when  $t = 0$  and is not present when  $t = 1$ . An alternative formulation of Eq. 2.15 is

$$l = bt_o, \quad (2.16)$$

where  $t_o$  is the observed transmittance, as opposed to the ideal transmittance  $t$ , and is defined as

$$t_o = (1 - \varepsilon)t + \varepsilon. \quad (2.17)$$

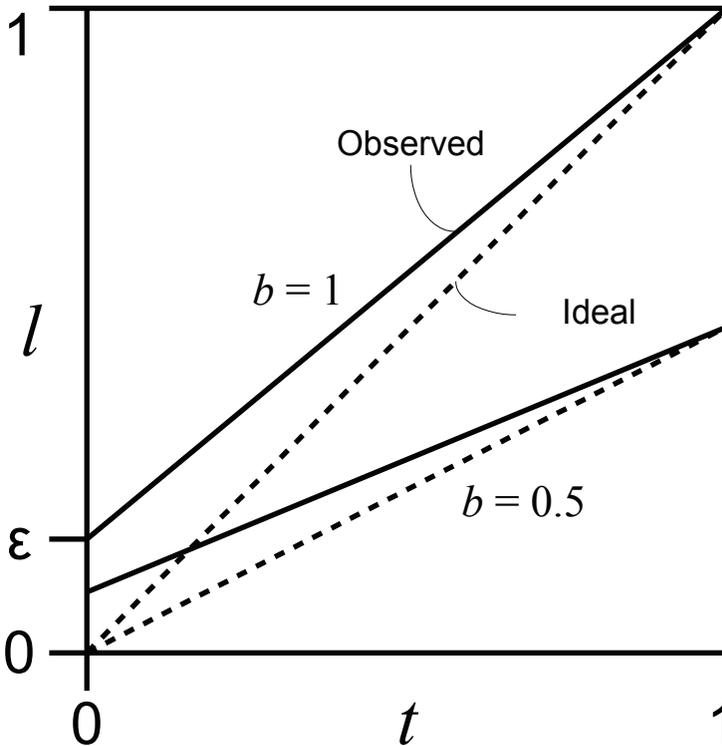
Figure 2.9 shows the linear model of leakage.

The leakage factor is dependent of the horizontal and vertical viewing angles  $\theta_H$  and  $\theta_V$ . Larger  $\theta_H$  and  $\theta_V$  mean larger  $\varepsilon$ . Typically, the leakage increase is different for the horizontal and vertical directions. This means that pixels have different  $\varepsilon$  for the viewer, depending on their position. For simplicity, unless differently stated, the value of  $\varepsilon$  is assumed to be constant across all the screen. The perception of leakage also depends on the ambient light: it can be seen more clearly in dark environments.

It should be noted that, in presence of multiple color components in the images or of multiple primary colors in the display, Eqs. 2.14- 2.17 apply to each of them separately.

### Backlight Diffusion

The LEDs in the backlight are normally driven by Pulse Width Modulation (PWM) signals. The response time of LEDs is in the order of microseconds, making it negligible in the case we consider. Consequently, the light intensity emitted by LED  $k$  is proportional to the duty-cycle



**Figure 2.9:** Linear modeling of leakage, which is proportional to the leakage factor  $\varepsilon$ , to the backlight intensity  $b$  and to  $(1 - t)$ , where  $t$  is the pixel transmittance (see Eq. 2.15).

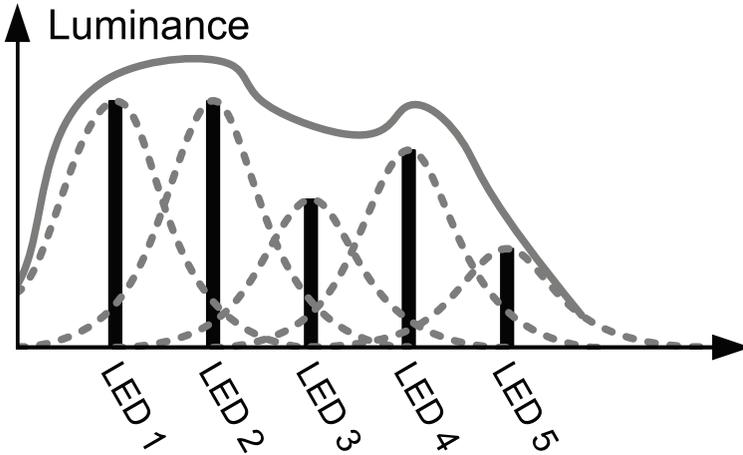
of the driving signal  $r_k$ . It is therefore possible to estimate the backlight power consumption  $p$  by averaging the duty-cycles:

$$p = \frac{\sum_{k=1}^M r_k}{M}, \quad (2.18)$$

where  $M$  is the number of LEDs / backlight segments.

The local luminance of the backlight results from the combination of the PSFs of all LEDs, each multiplied by the duty-cycle of the LED. The backlight  $b$  at a certain pixel is determined by

$$b = \sum_{k=1}^M h_k r_k, \quad (2.19)$$



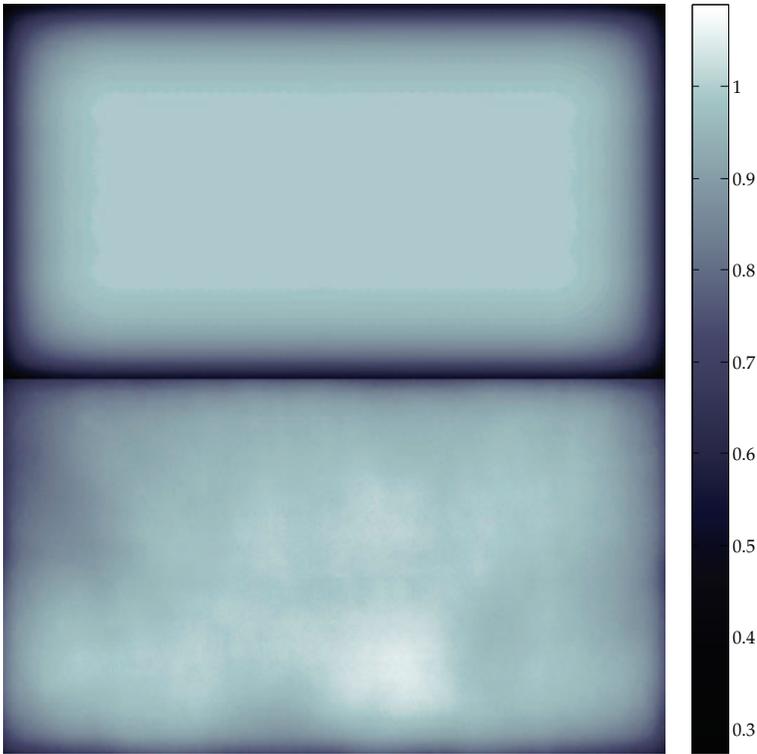
**Figure 2.10:** Example of PSF based light diffusion (section). The black bars represent the intensities of the LEDs, the dashed lines are the corresponding PSF (multiplied by the intensities) and the solid curve is the sum of all contributions.

where  $r_k$  is the duty-cycle, or intensity, of LED  $k$  and  $h_k$  is the value of the PSF of segment  $k$  at the pixel position. The values of  $r_k$  are normalized between 0 and 1, with 0 meaning that the LED is turned off and with 1 meaning that its light output is maximal. Each LED  $k$  contributes to  $b$  to the extent defined by the PSF, multiplied by the LED intensity  $r_k$ . The shape of the PSFs depends on the characteristics of the light source, light guide and light diffuser. Considering all pixels in the screen, Eq. 2.19 can be put in matrix form:

$$\mathbf{b} = \mathbf{H}\mathbf{r}, \quad (2.20)$$

where the column vector  $\mathbf{b}$  contains backlight values  $b$  for all pixels, the *influence matrix*  $\mathbf{H}$  ( $M$  columns and  $N$  rows, where  $N$  is the number of pixels in the screen) represents the PSFs, expressed by all the  $h$  values for all segments, and  $\mathbf{r}$  is a column vector with  $M$  LED values. A one-dimensional example of light diffusion based on PSF is shown in Figure 2.10.

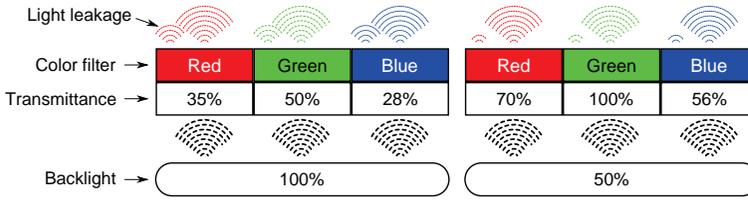
In real displays, the light distribution in the backlight when all LEDs are at full power is typically non uniform, as shown in the bottom part of Figure 2.11. Because of this, the maximum luminance achievable by different pixels might be different. This problem can be tackled by



**Figure 2.11:** Example of modeled (top) and measured (bottom) backlight. Here, the modeled backlight is obtained by applying a single PSF on many segments, causing dim edges and non-accurate representation of the actual backlight, one example of which is showed in the bottom part.

setting the *target peak luminance* to be some value below the *maximum peak luminance* achievable by the screen. The target peak luminance defines the luminance of the color white. By lowering it, more pixels can reach it in presence of non-uniformity. However, the level should not be set too low to avoid excessive reductions of brightness.

In the presence of many backlight segments, it might be impossible or unpractical to store precise measures of all the PSFs. To reduce the number of measurements on the screen or the amount of memory required for storage, it is possible to apply the same PSF on more than one segment. Particularly, this can be done for direct-lit backlights with many segments, which tend to have similar, although not equal, PSFs.



**Figure 2.12:** Example of brightness compensation; the full backlight (left) is dimmed to half power (right) and the RGB transmittances are correspondingly doubled. The lower backlight intensity has the positive effect of reducing light leakage. Excessive dimming can however cause clipping.

Edge-lit displays tend to have a lower number of segments and storing PSF data is less of a problem. The problem with using one PSF for many segments is that the segments close to the edges of the screen will trim it, resulting in a modeled backlight dimmer than the actual one because part of the PSF is actually neglected. Accuracy will generally be lower, as shown in Figure 2.11, where the modeled backlight resulting from one single PSF applied to a large number of segments (top) is compared to the real measured backlight (bottom). Reducing the target peak luminance can help to reduce differences between modeled and measured backlight.

### Brightness Compensation

Backlight dimming generally reduces the intensity of the backlight. The output luminance of the pixels is kept identical to that of the target luminance with the step of *brightness compensation*, also called brightness preservation [40] or pixel compensation [41]. The concept is illustrated in Figure 2.12.

Ideally, it is possible to calculate the compensated transmittance  $t_C$  from Eq. 2.14 by replacing the general luminance  $l$  with the target luminance  $l_y$ , as in

$$t_C = \frac{l_y}{b}. \quad (2.21)$$

In the practical case, the values of the observed physical luminance are restricted between  $\varepsilon$ , because of leakage, and 1, because higher values would mean that the LCs can amplify light, which is not true. Equation

2.21 is therefore so modified:

$$t_C = \begin{cases} 1 & \text{if } \frac{l_y}{b} > 1 \text{ (Condition I)} \\ 0 & \text{if } \frac{l_y}{b} < \varepsilon \text{ (Condition II)} \\ \frac{\frac{l_y}{b} - \varepsilon}{1 - \varepsilon} & \text{otherwise} \end{cases} . \quad (2.22)$$

If Condition I is verified, *clipping* occurs in the pixel, meaning that it is dimmer than desired. Condition II defines pixels that suffer from leakage and output a luminance higher than the target. Using Eq. 2.17 it is possible to calculate the observed physical compensated transmittance  $t_{Co}$ , if  $t$  is replaced with  $t_C$ :

$$t_{Co} = \begin{cases} 1 & \text{if } \frac{l_y}{b} > 1 \text{ (Condition I)} \\ \varepsilon & \text{if } \frac{l_y}{b} < \varepsilon \text{ (Condition II)} \\ \frac{l_y}{b} & \text{otherwise} \end{cases} . \quad (2.23)$$

An alternative notation for Eq. 2.23 is the following:

$$t_{Co} = \left[ \frac{l_y}{b} \right]_{\perp\varepsilon}^{\top 1} , \quad (2.24)$$

where  $\perp\varepsilon$  means that  $t_{Co} = \varepsilon$  if  $l_y/b < \varepsilon$  and  $\top 1$  means that  $t_{Co} = 1$  if  $l_y/b > 1$ . The kind of compensation in Eq. 2.22 is called hard clipping. Hard clipping compensates the loss of luminance as much as possible and minimizes the physical luminance error. However, it can cause annoying posterization artifacts by flattening some areas of the image. Soft clipping is an alternative approach that operates a less aggressive compensation for high target luminance, with the effect of reducing posterization and potentially improving the perceptual appearance [40].

Equations 2.14-2.22, which have been presented in the previous sections, apply to monochrome signals. In presence of more color components, e.g. RGB, they are to be applied on each of them. This may introduce a new artifact caused by hard clipping, that is color distortion occurring when one of the RGB sub-pixels clips but not the others. A way to combat this problem could be to compensate the three channels together and maintain the transmittance ratios. This will result in lower luminance but less color shifts. In this thesis, unless otherwise noted, hard clipping with independent RGB compensation is assumed.

### Human Perception of Luminance

The HVS has a non-linear response to luminance, roughly logarithmic. As absolute luminance increases, the sensitivity to variations of luminance in the HVS decreases. In general, the perceptual response can be expressed as

$$l_U = g(l), \quad (2.25)$$

where  $g$  is a non-decreasing function of the physical luminance  $l$ ;  $l_U$  is called the perceived luminance and is said to be in the perceptual domain, as opposed to the physical domain where  $l$  is expressed. In general, equal steps of  $l_U$  correspond to varying steps in  $l$  and vice versa. For luminance in the range of 1-100 cd/m<sup>2</sup>,  $g$  can be approximated with a gamma function [42]:

$$l_U = l^{\frac{1}{\gamma}}, \quad (2.26)$$

Values of  $\gamma$  between 2.2 and 3.0 have been reported to obtain excellent perceptual performance [17]. A  $\gamma$  of 2.2 approximates very closely the gamma of the sRGB color space [39], which makes it coincide with the perceptual domain.

For displays with higher luminance, other functions should be used to model the perceptual response. Aydın et. al. have defined experimentally a function mapping physical and perceived luminance in the range 1-10<sup>6</sup> cd/m<sup>2</sup>. The subjective data has been fitted with a logarithmic function in [2], resulting in the following equation,

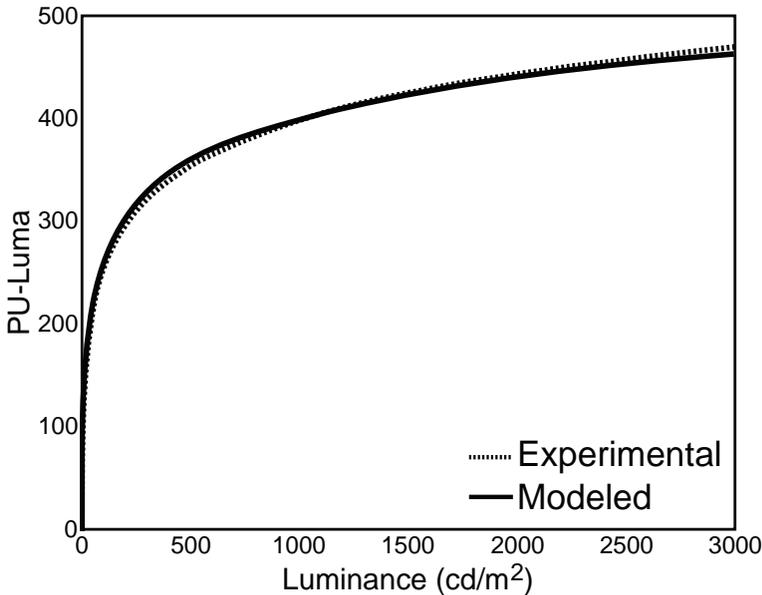
$$l_U = 66.25 \times \log_e(0.56 \times l^{0.88} + 1), \quad (2.27)$$

which is plotted in Figure 2.13. Unless differently stated, in this work  $l_U$  is modeled using Eq. 2.26 with  $\gamma = 2.2$ .

### Quality Assessment

The objective quality of images produced by backlight dimming algorithms can be measured inside the model, for example in terms of the difference between the target image  $\mathbf{y}$  and the output  $\mathbf{x}$  of the algorithm in the physical or in the perceptual domain. Common measures include the Mean Absolute Error (MAE):

$$\text{MAE} = \frac{\sum_{i=1}^N |y_i - x_i|}{N}, \quad (2.28)$$



**Figure 2.13:** Function mapping physical luminance for high peak luminance displays to PU luminance. The experimental data is taken from [42].

the Mean Squared Error (MSE):

$$\text{MAE} = \frac{\sum_{i=1}^N (y_i - x_i)^2}{N}, \quad (2.29)$$

and the PSNR, which is built on MSE:

$$\text{PSNR} = 10 \times \log_{10} \left( \frac{\text{MAX}^2}{\text{MSE}} \right), \quad (2.30)$$

where  $N$  is the number of pixels in the screen and MAX is the maximum pixel value. For non-normalized 8-bit sRGB values,  $\text{MAX} = 255$ . In this work normalized pixel values are normally used, therefore  $\text{MAX} = 1$  unless differently stated. Other quality metrics can of course be used.

**LabPSNR** LabPSNR is a quality metric based on PSNR using the CIE 1976  $L^*a^*b^*$  color space (from here on referred as “Lab” for brevity) instead of sRGB. The reason is that Lab was designed to be perceptually uniform regarding color distortion, while sRGB was not. This means

that a change of magnitude in Lab approximates an equivalent change in the perceived color. Conversion from linearized RGB values to Lab is done through the intermediate XYZ color space as follows [39], [43]:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.412 & 0.358 & 0.180 \\ 0.213 & 0.715 & 0.072 \\ 0.019 & 0.119 & 0.950 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (2.31)$$

$$\begin{aligned} L^* &= 116 \cdot f(Y/Y_n) - 16 \\ a^* &= 500 \cdot [f(X/X_n) - f(Y/Y_n)] \\ b^* &= 200 \cdot [f(Y/Y_n) - f(Z/Z_n)] \end{aligned}, \quad (2.32)$$

where  $f(k) = \begin{cases} k^{1/3} & \text{if } k > 0.008856 \\ 7.787 \cdot k + \frac{16}{116} & \text{otherwise} \end{cases}$

where  $X_n$ ,  $Y_n$  and  $Z_n$  are the X, Y and Z values for the reference white, respectively. The color difference  $\Delta E$ , considering both luminance and chrominance difference, is defined as

$$\Delta E = \sqrt{\Delta L^{*2} + \Delta a^{*2} + \Delta b^{*2}}, \quad (2.33)$$

where  $\Delta L^*$ ,  $\Delta a^*$  and  $\Delta b^*$  define the differences between the target and output pixel measured for  $L^*$ ,  $a^*$  and  $b^*$  components. LabPSNR is defined by taking the definition of PSNR (Eq. 2.30) and replacing MSE with mean squared  $\Delta E$ :

$$LabPSNR = 10 \cdot \log_{10} \left( \frac{(\Delta E_{max})^2}{\frac{1}{N} \sum_{i=1}^N \Delta E_i^2} \right), \quad (2.34)$$

where  $\Delta E_i$  is the color difference at pixel  $i$ , determined by Eq. 2.33 and  $\Delta E_{max}$  is the difference between black and white (normalized sRGB triplets (0,0,0) and (1,1,1)) in the Lab color space: its value is 100.

## Equipment

This section presents three LCDs with local backlight that have been modeled and used in experiments.

The first screen is a direct-lit panel with 221 backlight segments, organized in 13 rows and 17 columns. The PSF, equal for all segments, is modeled with a bi-dimensional Gaussian function.

The second screen is the model of a real 47 inch Full High Definition (HD) display manufactured by SIM2 [44]. The backlight has 2202 segments placed in a hexagonal grid. As in the previous screen, all segments use the same PSF, provided by the manufacturer. This was done for practical reasons, however the model allows to assign a different PSF to each segment. An example of backlight generated by this screen is shown in the bottom-left corner of Figure 2.7.

The third screen is edge-lit and has 16 segments, placed in 8 rows and 2 columns. As the previous display, this one is also modeled on a real device. However, this display uses a specific PSF for each LED, obtained through measurements. An example of backlight generated by this screen is shown in the bottom-right corner of Figure 2.7

All screens have a default resolution of  $1920 \times 1080$  pixels, but it is possible to define a downscaling factor that resizes the displays and the PSFs accordingly. Different factors can be specified for the horizontal and for the vertical dimension.

### 2.3.3 Optimization Based Algorithms

This section presents backlight dimming algorithms based on optimization that have been developed during the Ph. D. project object of this thesis. The target of optimization is to find the best trade-off between clipping and leakage and to reduce the power consumption in local dimming systems. The first algorithm was presented in [13] and was extended in [3], [5], [10]. Proposals for reducing the complexity of the method have been brought forward in [3], [10], [13]. All sections report experimental results assessing the performance of the algorithms.

#### Original Formulation

In [13], backlight dimming is modeled as an optimization problem. The goal is the minimization of a cost function defined by the norm of the difference between the target physical luminance  $\mathbf{y}$  and the actual physical luminance  $\mathbf{x}$  emitted by the display pixels to obtain the maximum quality. The reduction of power consumption is a by-product of quality

optimization, due to the fact that leakage is included in the optimization, which pushes to reduce the backlight intensity. The output  $\mathbf{x}$  is determined by the backlight  $\mathbf{b}$  and by the pixel transmittances  $\mathbf{t}$  (Eq. 2.16). The backlight  $\mathbf{b}$ , calculated as in Eq. 2.20, depends on the influence matrix  $\mathbf{H}$  and on the LED values  $\mathbf{r}$ . The values of  $\mathbf{t}$  are within the leakage factor  $\varepsilon$  and 1, while those of  $\mathbf{r}$  are limited between 0 and 1.

The formulation of the problem is the following:

$$\begin{aligned}
 & \text{minimize} && \|(\mathbf{y} - \mathbf{x})\| \\
 & \text{subject to} && \mathbf{x} = \mathbf{b} \circ \mathbf{t} \\
 & && \mathbf{b} = \mathbf{H}\mathbf{r} \\
 & && \varepsilon \leq t_i \leq 1, \quad k = 1, \dots, N \\
 & && 0 \leq r_k \leq 1, \quad k = 1, \dots, M
 \end{aligned} \tag{2.35}$$

where the  $\circ$  operator defines element-wise multiplication. The norm to be minimized can be any norm, for example  $\ell_1$  or  $\ell_2$ . The optimal  $r_k$  can be found using software solvers taking the problem and data as input [45], [46]. After optimization, the values of  $\mathbf{r}$  result in the optimal backlight. Problem 2.35 can be reformulated in convex form as follows:

$$\begin{aligned}
 & \text{minimize} && \|\boldsymbol{\lambda}\| \\
 & \text{subject to} && \mathbf{b} = \mathbf{H}\mathbf{r} \\
 & && \boldsymbol{\lambda} \geq (\varepsilon \circ \mathbf{b}) - \mathbf{y} \\
 & && \boldsymbol{\lambda} \geq \mathbf{y} - \mathbf{b} \\
 & && \boldsymbol{\lambda} \geq 0 \\
 & && 0 \leq r_k \leq 1, \quad k = 1, \dots, M
 \end{aligned} \tag{2.36}$$

where  $\boldsymbol{\lambda}$  is an auxiliary vector containing bounds for the difference between target and output pixels. The reader may consult [13] to find more information about the conversion from Problem 2.35 to Problem 2.36.

To make a simple assessment of its performance, the optimization-based method has been compared against the clipper-free and generally good quality algorithm from [36]. The norm was minimized in  $\ell_2$ . The algorithms have calculated the backlight for the edge-lit screen described in Section 2.3.2 with  $\varepsilon = 0.0002$ . The display has been downsampled by a factor of 8, horizontally and vertically. Both algorithms have been implemented in Matlab. The solver used for the optimization based approach is provided by the CVX package [45], [46].

	Optimal $\ell_2$	Albrecht et al. [36]
MSE	7.319 E-9	8.420 E-9
Power	0.7376	0.7506

**Table 2.1:** Comparison of the basic optimization approach minimizing  $\ell_2$  norm against another algorithm.

Eight grayscale images were used in the experiment, shown in Figure B.2 in Appendix B. The set of chosen images includes several image types like synthetic graphics, natural images and compressed pictures. The MSE has been measured on the luma channel in the physical domain at the downscaled resolution. Table 2.1 reports the average MSE and power consumption.

The optimization algorithm can achieve at the same time lower physical MSE and lower power consumption. The reason for this is that leakage is included in the optimization, while it is neglected in [36]. Further results can be found in [13].

## Extensions

Optimization Problem 2.35 has two limits. One limit is that the optimization is done in the physical domain, not in the perceptual, which results in sub-optimal perceived results. The other limit is that the cost function only considers the norm and, while power savings result from including leakage in the optimization, it is not possible to find the optimal backlight level for lower levels of energy. To address these problems, two extensions to Problem 2.35 have been proposed in [3] and [5]. The first extension is the introduction of an error weighting matrix  $\mathbf{w}$  that can assign specific weights to the pixel differences between  $\mathbf{y}$  and  $\mathbf{x}$ . The second extension is the inclusion of an adjustable term penalizing power consumption in the cost function.

The weighting matrix can be used to approximate a better perceptual result from the optimization. In the physical domain, the luminance error at pixel  $i$  between  $y_i$  and  $x_i$  is only determined by the difference  $y_i - x_i$ . However, in the perceptual domain this difference might be more or less noticeable due to the non-linear response to luminance of the HVS. As discussed in Section 2.3.2, the perceptual sensitivity to luminance variations decreases as the absolute luminance increases.

Following this, leakage errors are more evident than clipping errors of the same physical intensity. The pixel-specific error weights can be used to give more emphasis to errors in the low luminance range, following the perceptual response. If this is  $g$  (as in Eq. 2.25), the proposal is to use, as a weight  $w_i$  for pixel  $i$ , the slope of the curve at the target luminance  $y_i$ :

$$w_i = g'(y_i). \quad (2.37)$$

If  $g$  is modeled using a gamma function as in Eq. 2.26, then Eq. 2.37 becomes

$$w_i = \frac{1}{\gamma} \times y^{1-\frac{1}{\gamma}}. \quad (2.38)$$

Power consumption can be penalized by including a penalty term in the cost function to be minimized. Power consumption  $p$  can be estimated from the LED values  $r_k$  as in Eq. 2.18. The impact of  $p$  can be adjusted by multiplying it by the coefficient  $q$ , called the power weight or power penalty. Larger values of  $q$  favor backlights with lower intensity.

To include these two extensions, Problem 2.35 is modified as follows:

$$\begin{aligned} & \text{minimize} && \|(\mathbf{y} - \mathbf{x}) \circ \mathbf{w}\| + q \times p \\ & \text{subject to} && \mathbf{x} = \mathbf{b} \circ \mathbf{t} \\ & && \mathbf{b} = \mathbf{H}\mathbf{r} \\ & && \varepsilon \leq t_i \leq 1, \quad i = 1, \dots, N \\ & && 0 \leq r_k \leq 1, \quad k = 1, \dots, M \end{aligned} \quad (2.39)$$

The linear reformulation consequently becomes:

$$\begin{aligned} & \text{minimize} && \|\boldsymbol{\lambda}\| + q \times p \\ & \text{subject to} && \mathbf{b} = \mathbf{H}\mathbf{r} \\ & && \boldsymbol{\lambda} \geq ((\boldsymbol{\varepsilon} \circ \mathbf{b}) - \mathbf{y}) \circ \mathbf{w} \\ & && \boldsymbol{\lambda} \geq (\mathbf{y} - \mathbf{b}) \circ \mathbf{w} \\ & && \boldsymbol{\lambda} \geq 0 \\ & && 0 \leq r_k \leq 1, \quad k = 1, \dots, M \end{aligned} \quad (2.40)$$

It can be noticed that if  $\mathbf{w}$  is set as a vector of ones and  $q$  is set equal to zero, Problem 2.39 is equivalent to Problem 2.35 and Problem 2.40 to Problem 2.36.

Table 2.2 shows how the MSE increases and power consumption decreases when  $q$  is increased. The data is generated from simulations

$q$	0	1	5
MSE	7.319 E-9	9.293 E-8	6.284 E-6
Power	0.7376	0.6535	0.5395

**Table 2.2:** Increase of physical MSE and reduction of power consumption as  $q$  increases in  $\ell_2$  optimization with no perceptual error weighting (Problem 2.39).

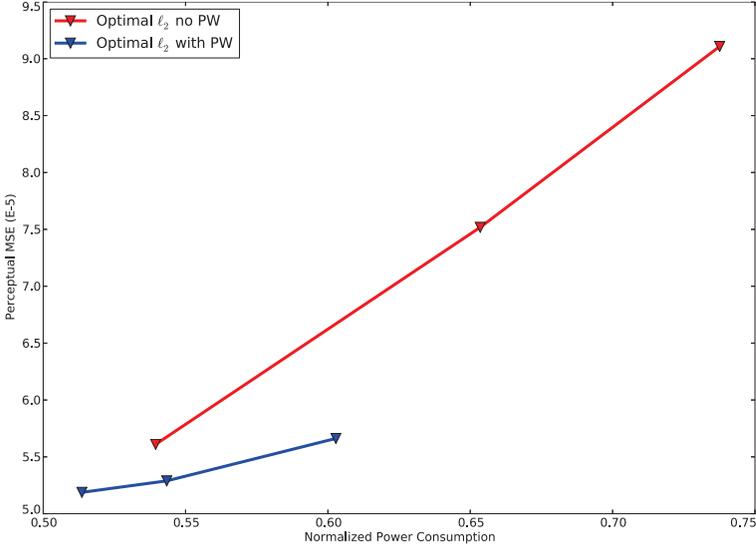
having the same parameters as that of the previous section (physical  $\ell_2$  optimization, edge-lit screen with  $\varepsilon = 0.0002$  and downscaled by a factor 8, over 8 grayscale images), except that they are based on Problem 2.39 with no perceptual weighting and varying values of  $q$ . Power consumption is clearly reduced, but at the same time the error increases. In general, it appears that higher quality solutions require more energy. The exception to this is leakage, which is reduced by dimmer backlights.

If MSE is measured in the perceptual domain, the error is larger than in Table 2.2, as shown in Figure 2.14. Interestingly, however, increasing the value of  $q$  reduces the error, because leakage error is reduced by the lower backlight intensity. Enabling perceptual error weighting (as described previously) brings MSE even lower; the penalization of leakage over clipping also helps reducing power consumption further.

In [5], the power penalization mechanism has been tested more extensively. Figure 2.15 compares the results of solving 2.39 for  $\ell_1$  optimization with perceptual error weighting for the SIM2 display described in Section 2.3.2, with  $\varepsilon = 0.001$  and a downscaling factor of 4. Eight grayscale images were used, shown in Figure B.2 in Appendix B. Results for the backlight dimming algorithms presented in [36], [28], [26] and the Max algorithm (see Section 2.3.1) are reported for comparison. PSNR is measured in the perceptual domain. The optimization based approach achieves higher average PSNR than other algorithms at a given power level. Further results can be found in [5] and [3], available in Appendix A.

## Color

Optimization Problem 2.39 only minimizes the cost function over one color channel. The formulation has been generalized in [10] to allow



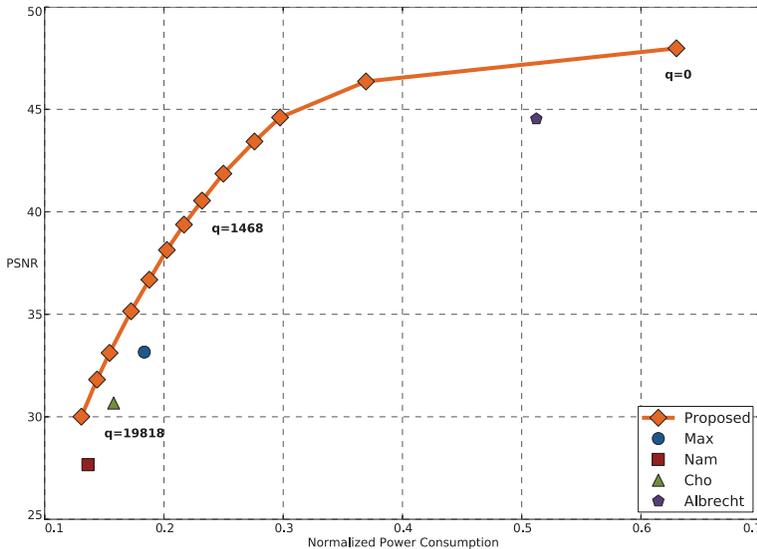
**Figure 2.14:** Performance of perceptual weighting (PW) of the error; when enabled, MSE can be reduced in the perceptual domain.

optimization in multiple color channel:

$$\begin{aligned}
 & \text{minimize} && \left\| \sum_{j=R,G,B} (\mathbf{y}_j - \mathbf{x}_j) \circ \mathbf{w}_j \right\| + q \times p \\
 & \text{subject to} && \mathbf{x}_j = \mathbf{b} \circ \mathbf{t}_j; \quad j = R, G, B \\
 & && \mathbf{b} = \mathbf{H}\mathbf{r} \\
 & && \varepsilon \leq t_{ij} \leq 1; \quad i = 1, \dots, N; j = R, G, B \\
 & && 0 \leq r_k \leq 1; \quad k = 1, \dots, M
 \end{aligned} \tag{2.41}$$

Including color improves the quality of the result but also increases complexity, since the number of variables and constraints is tripled. While RGB components are assumed, the generalized formulation can be adapted to a different set of primaries. The three RGB sub-pixels have different transmittances but share the same backlight intensity. A specific error weight vector can be assigned to each color channel.

Table 2.3 compares single-channel optimization (Problem 2.39) to the color aware formulation (Problem 2.41). The single-channel results are the same as in the previous section (without perceptual weighting of the error), but this time the backlight is applied on the color image and MSE is measured on it. Although it increases complexity, optimizing



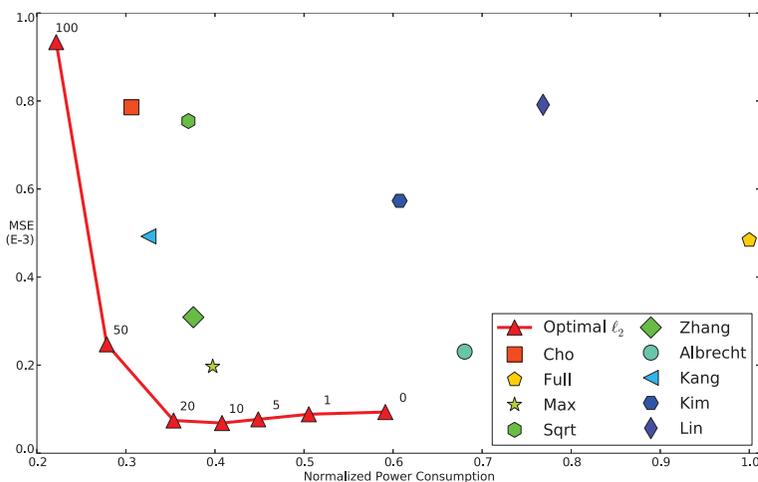
**Figure 2.15:** Comparison of the optimization based approach with  $\ell_1$  minimization, perceptual error weighting and penalization of power consumption against other algorithms. References are: Max [25], Nam [28], Cho [26], Albrecht [36].

Luma	$q$	0	1	5
	MSE	1.027 E-3	2.636 E-3	2.964 E-3
	Power	0.7376	0.6535	0.5395
RGB	$q$	0	1	5
	MSE	7.671 E-9	2.877 E-8	1.306 E-6
	Power	0.8440	0.7798	0.6837

**Table 2.3:** Optimization over the luma component only (top) compared to that over RGB components (bottom).

over color helps to reduce the error significantly.

Another simulation, reported in [10], compares the optimization of Problem 2.41 against other algorithms: [26], [27], [29], [31], [35], [36], the Max, Avg and Sqrt algorithms. Optimization is done on the direct-lit display and on the 8 color images shown in Figure B.1 in Appendix B, with  $\varepsilon = 0.001$  at downscaling 5. The resulting backlight is applied at the original Full HD resolution (1920x1080) and the MSE is measured



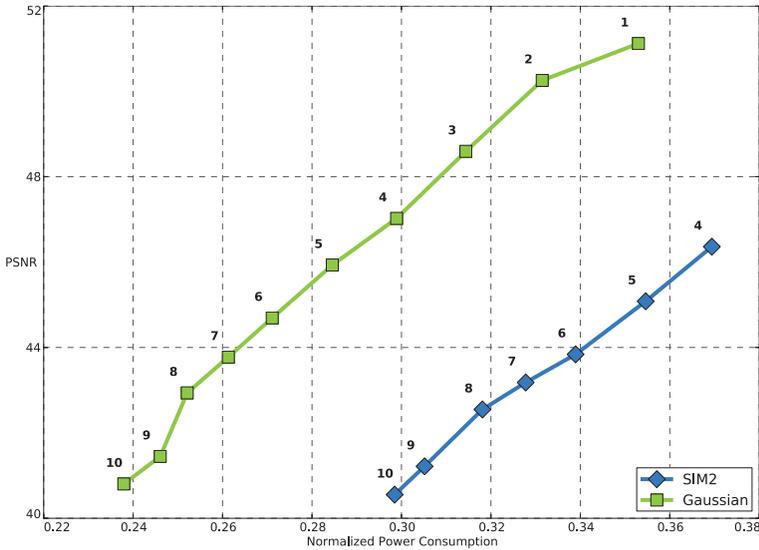
**Figure 2.16:** Comparison of the optimization based approach with  $\ell_2$  minimization, perceptual error weighting and penalization of power consumption against other algorithms. Results are for a direct-lit screen with 2202 segments, with  $\varepsilon = 0.001$  and downscaling factor 5. Labels indicate power penalty. References are: Max and Sqrt [25], Cho [26], Zhang [27], Albrecht [36], Kang [31], Kim [29], Lin [35].

on all color components in the perceptual domain. Again, this approach achieves the lowest error at a given power level.

### Reduction of Complexity

The optimization-based algorithms presented in Section 2.3.3 can obtain high quality results but suffers from high complexity. For instance, Problems 2.35 and 2.39 have millions of variables and constraints in the case of a Full HD image. For Problem 2.41 the amount is three times as much. The following paragraphs discuss strategies aimed at reducing the complexity of these algorithms.

**Downscaling** One solution is to downscale the input image and downscale by the same factor the screen and the PSFs in the model, minimize the cost function on the downscaled image and then apply the result on the original full resolution image. It was shown, in [5], that downscaling reduces quality and power consumption. In that case, bicubic downscaling was performed on the image and bilinear downscaling to the PSFs.



**Figure 2.17:** Impact of downscaling on PSNR and power consumption for two screens. The labels indicate the downscaling factors.

Figure 2.17 shows the impact of the downscaling factor over the Gaussian screen and the SIM2 screen presented in Section 2.3.2; the factor is varied from 1 to 10 for the first screen and from 4 to 10 for the second one. Optimization was done with Problem 2.39 for  $\ell_1$  with perceptual error weighting and  $\varepsilon = 0.001$ . The average PSNR, measured on the 8 images were shown in Figure B.2 in Appendix B, is measured in the perceptual domain.

The Gaussian screen seems to have better performance compared to the other screen. While it is worth noting that this screen is idealized and not modeled on a real display, while the other is, the reason for this is due to the PSFs, which are smaller than those of the direct-lit screen and therefore need to compromise less in case leaking and clipped pixels are in the same area.

The problem with downscaling is that it applies a low pass filter to the target image, blurring very bright or very dark pixels, and therefore causing a larger number of pixels to be clipped or leaking.

**Block-wise Uniform Backlight** Complexity can also be reduced by introducing in the model approximations that decrease the amount of calculations. In [13], the authors suggest to divide the backlight in blocks and assume that the intensity inside them is constant. This is a reasonable assumption as long as the blocks have a relatively small size, since the backlight resolution is much lower than pixel resolution and since light diffuses smoothly. This allows to reduce the number of variables needed to represent the backlight, e.g. if the block size is  $10 \times 10$ , the number of variables is reduced 100 times. It is still necessary to have one variable for each pixel transmittance; however, since the backlight is constant inside the block, pixels with the same value will have the same error. These pixels with the same value can be represented with a single variable, but with a larger weight in the error, equal to the number of pixels having the same value. Constraints  $\lambda_i \geq y_i - b_i$  from Problem 2.36, where  $i$  refers to a pixel in the backlight block, becomes

$$\lambda'_v \geq n_v(v - b_B), \quad (2.42)$$

where  $n_v$  is the number of pixels with value  $v$  inside the block, and  $b_B$  is the backlight intensity inside the block. Constraint  $\lambda_i \geq \varepsilon b_i - y_i$  similarly becomes

$$\lambda'_v \geq n_v(\varepsilon b_B - v). \quad (2.43)$$

The authors suggest that, for this second constraint, the backlight blocks can be even larger because  $\varepsilon$  values are usually very small, making the variation of  $\varepsilon b$  very smooth.

In the worst case, that is when all possible pixel values appear at least once inside the backlight block, the number of constraints for the block will be equal to the number of values (e.g. 256 in a 8-bit image).

**Sub-sampling Using Only Distorted Pixels** Another approach that can reduce complexity is sub-sampling the target image and finding the optimal solution on the pixel subset only. It was shown that it is possible to obtain nearly-optimal results by using about 10-25% of the input pixels [3]. It should be noted that sub-sampling differs from downscaling, because the sub-sampling takes a subset of the original pixels (samples) without processing them.

In [3], it is noted that the error between the target  $\mathbf{y}$  and the display output  $\mathbf{x}$  is due to distorted pixels, that is clipped and leaking pixels.

The rest of the pixels do not contribute to the error because this transmittance can compensate the dimmed backlight. This indicates that distorted pixels are those that influence the optimal result the most. Therefore, solving the optimization problem on these pixels should only result in a backlight reasonably close to the optimum.

In order to validate this proposal, it is necessary to find a subset of the target pixels to optimize on. This is done by applying an initial backlight (calculated with any algorithm) on the full set of pixels and selecting the subset of pixels that are distorted. Optimization is then performed on this subset and the resulting backlight is applied to the full set. If new pixels are distorted, and were not distorted with the previous backlight, they are added to the subset. This process is re-iterated until no new pixel is distorted.

Optimizing on a subset requires less time than optimizing on the complete set, but implies many iterations, which might result in a longer overall time. However, the goal is to show that optimal or nearly optimal solutions can be found by optimizing on a subset of the target pixels. Processing time is therefore not critical at this stage.

The cost function based on the distorted pixels only can be expressed, in the case of  $\ell_1$  norm, as

$$f = \sum_{i \in D_C} (y_i - b_i) + \sum_{i \in D_L} (\varepsilon b_i - y_i), \quad (2.44)$$

where  $D_C$  is the set of clipped pixels and  $D_L$  is the set of leaking pixels.

This approach has been tested on two displays (the edge-lit and SIM2 screen described in Section 2.3.2), with downscaling 4 and 10. Optimization was done in  $\ell_1$  and  $\ell_2$  with  $\varepsilon = 0.001$  on the luma component of the 24 images of the Kodak dataset, shown in Figure B.4 in Appendix B. Table 2.4 reports the comparison with optimization done on the full set. MAE and MSE have been measured in the physical domain at the downscaled resolution; power consumption is reported as well. The initial solution was calculated using the Max algorithm [25].

In all cases, the subset strategy is successful and provides results virtually identical to those of full set optimization, using only between 9% and 26% of the original pixels, depending on the screen. It should be noted that power consumption tends to increase. More results can be found in [3], available in Appendix A.

	Edge		SIM2	
	Original	Subset	Original	Subset
<b>Avg. LED value</b>	82.36%	82.89%	73.36%	67.12%
<b>Avg. MAE</b>	3.78 E-5	3.78 E-5	1.05 E-5	1.05 E-5
<b>Avg. iterations</b>	-	2.59	-	2.67
<b>Sub-sampling % in last iteration</b>	-	9.70%	-	25.70%

$\ell_1$  optimization, downscaled by 10

	Edge		SIM2	
	Original	Subset	Original	Subset
<b>Avg. LED value</b>	87.91%	89.30%	81.21%	75.77%
<b>Avg. MAE</b>	4.62 E-5	4.62 E-5	1.81 E-5	1.81 E-5
<b>Avg. iterations</b>	-	2.69	-	2.91
<b>Sub-sampling % in last iteration</b>	-	9.55%	-	17.73%

$\ell_1$  optimization, downscaled by 4

	Edge		SIM2	
	Original	Subset	Original	Subset
<b>Avg. LED value</b>	76.50%	84.66%	48.80%	61.03%
<b>Avg. MSE</b>	4.04 E-8	4.04 E-8	8.70 E-9	8.70 E-9
<b>Avg. iterations</b>	-	2.59	-	2.59
<b>Sub-sampling % in last iteration</b>	-	9.70%	-	25.86%

$\ell_2$  optimization, downscaled by 10

	Edge		SIM2	
	Original	Subset	Original	Subset
<b>Avg. LED value</b>	87.82%	92.62%	57.72%	67.51%
<b>Avg. MSE</b>	5.73 E-8	5.73 E-8	1.48 E-7	1.48 E-7
<b>Avg. iterations</b>	-	2.72	-	2.97
<b>Sub-sampling % in last iteration</b>	-	9.48%	-	17.95%

$\ell_2$  optimization, downscaled by 4

**Table 2.4:** Optimization on a subset of pixels can still achieve optimal results.

**Min-Max Color Optimization** A method to reduce the complexity of Problem 2.41 has been proposed in [10]. The proposal approximates the optimal backlight for color images by using only a third of the variables.

In a RGB pixels, there are three sub-pixels which can contribute to the error. In general, the three values are different and it is possible to

find the maximum and the minimum of the three. The linear formulation of the single color channel optimization (Problem 2.36) specifies an upper and a lower threshold for the variables  $\lambda_i$  to be minimized. In a RGB pixel, the sub-pixel with the minimum value is the first one that could present leakage, while the sub-pixel with the maximum value is the first that could be hit by clipping. The idea is that a backlight intensity that minimizes the error considering only the minimum and maximum sub-pixels, then the median sub-pixel is probably able to compensate this intensity and produce no error. The constraints in Problem 2.36 can be modified by replacing  $\mathbf{y}$  with  $\mathbf{y}_{min}$  or  $\mathbf{y}_{max}$ , where  $\mathbf{y}_{min}$  contains the minimum sub-pixel values and  $\mathbf{y}_{max}$  the maximum sub-pixel values:

$$\begin{aligned}
 & \text{minimize} && \|\boldsymbol{\lambda}\| + q \times p \\
 & \text{subject to} && \mathbf{b} = \mathbf{H}\mathbf{r} \\
 & && \boldsymbol{\lambda} \geq ((\boldsymbol{\varepsilon} \circ \mathbf{b}) - \mathbf{y}_{min}) \circ \mathbf{w}_{min} \\
 & && \boldsymbol{\lambda} \geq (\mathbf{y}_{max} - \mathbf{b}) \circ \mathbf{w}_{max} \\
 & && \boldsymbol{\lambda} \geq 0 \\
 & && 0 \leq r_k \leq 1, \quad k = 1, \dots, M
 \end{aligned} \tag{2.45}$$

This reduces the  $\lambda_i$  variables to one third of its original size.

The effectiveness of this approach has been tested against optimization for the full color Problem 2.41. Optimization was done on the edge-lit and direct-lit SIM2 displays, on 32 color images (shown in Figures B.1 and B.3 in Appendix B), with  $\varepsilon = 0.001$  and  $\varepsilon = 0.0002$  for both  $\ell_1$  and  $\ell_2$ ; power penalty  $q$  was set to zero and several downscaling factors were used (8 and 10 for the SIM2, 4, 5, 6 and 8 for the edge-lit).

Table 2.5 shows the results for the edge-lit screen (downscaling factor 4) and the SIM2 screen (downscaling factor 10). In  $\ell_2$  minimization, MSE increases at most by 7.84%. Power consumption increases slightly increases with values between 0.68 and 2.56. Results for other downscaling factors and for  $\ell_1$  minimization are similar and not reported here. These results support the statement that optimization done using only the minimum and maximum sub-pixels as constraints can achieve good results, close to the optimum, while reducing variables to a third. For more data consult [10].

	$\varepsilon = 0.0002$		$\varepsilon = 0.001$	
	Power	MSE	Power	MSE
Full color	74.47	2.731 E-4	57.28	9.202 E-4
Color Min-Max	76.87	2.820 E-4	59.84	9.574 E-4
Variation	2.40	3.26%	2.56	4.04%

	$\varepsilon = 0.0002$		$\varepsilon = 0.001$	
	Power	MSE	Power	MSE
Full color	38.55	4.584 E-5	31.78	1.247 E-4
Color Min-Max	39.30	4.807 E-5	32.46	1.345 E-4
Variation	0.75	4.86%	0.68	7.84%

**Table 2.5:** Comparison of full and reduced complexity color  $\ell_2$  minimization on edge-lit (top) and direct-lit (bottom) panels.

### 2.3.4 Gradient Descent

This section presents another backlight dimming algorithm based on iterative search by Gradient Descent (GD), originally presented in [3] and improved in [10]. An approach to reduce its complexity is also presented.

#### Background

Given a display with  $M$  local backlight segments, the set of LED intensities  $r_k$  (with  $k = 1, \dots, M$ ) can be considered as the coordinate of a specific backlight in the solution space. The solution space has  $M$  dimensions and, for all possible backlights on the screen, the cost function  $f$  associates a cost to each of them. Considering a point/backlight in the solution space, it is possible to calculate the gradient of the cost function, that is the set of its partial derivatives over all dimensions, or to estimate it numerically if there is no closed form. The gradient indicates in which direction the cost increases the most from the current solution, in terms of variation of each LED value  $r_k$ . Moving in the opposite direction will instead produce the fastest cost decrease. If a step is taken in that direction, it is possible to find a new set of LED values having a lower cost. This new solution can be used to calculate a new gradient and so on, iterating as desired. At iteration  $j + 1$ , the solution

is calculated by

$$\mathbf{r}_{j+1} = \mathbf{r}_j - s\nabla f(\mathbf{r}_j), \quad (2.46)$$

where  $\mathbf{r}_j$  is the solution for iteration  $j$ ,  $f$  is the cost function for  $\mathbf{y}$ ,  $\nabla$  is the gradient operator and  $s$  the step size. The minus sign is due to the fact that the goal is cost minimization, therefore the step should be taken in the direction opposite to the gradient. The size of  $s$  can be chosen adaptively and be different in each iteration. Short steps guarantee a decrease in cost but require more iterations before the optimum is reached, while large steps can speed up the process but risk to deviate from the optimum. Finding the best or a good enough step size can be an optimization problem inside the optimization problem, and several approaches can be used. In general, it can be acceptable to find a step that yields a sufficient decrease in cost; in other cases it might be necessary to find the step that produces the maximum cost reduction, at the cost of higher complexity.

Compared to the optimization based approach from the previous section, this method allows to find minimum points for all cost differentiable functions, even in the perceptual domain. It is also very flexible, because it can generally improve any solution. In presence of a low number of segments, gradient can be calculated rather quickly, resulting in fast iterations.

## Experiments

The GD algorithm was tested in a series of experiments reported in [10]. The optimal backlight for  $\ell_1$  and  $\ell_2$  is calculated on the edge-lit screen with  $\varepsilon = 0.001$  for 32 color images (shown in Figures B.1 and B.3 in Appendix B). The starting solution is provided by the Min-Max subpixel approach detailed in Section 2.3.3, minimizing the same norm at a downscaling factor of 5, but the GD runs at full resolution. The cost function is the same as that of Problem 2.41 but in the perceptual domain and without perceptual weighting of the error. The power penalty  $q$  was set to several values in order to obtain solutions at several levels of power consumption to enable a fair comparison with other algorithms.

In each iteration of the GD, the step  $s$  was found through golden section search, a bisection method that allows to find a bracketed minimum of a function iteratively and with a given precision, ensuring fast con-

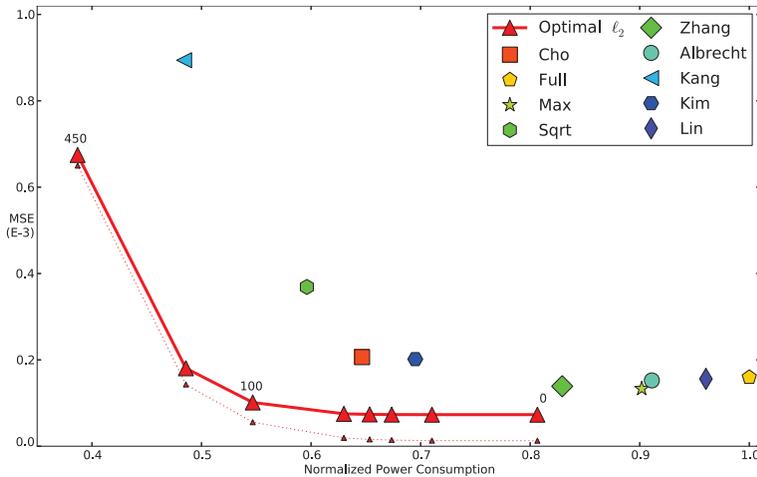
vergence and reducing the number of function evaluations [47]. Given the current solution ( $\mathbf{r}_i$ ) and its gradient, the cost of the next solution ( $\mathbf{r}_{i+1}$ ) is a function of  $s$ . Golden ratio search is applied on this function, and the initial bracketing is given by  $s = 0$  and the largest value of  $s$  ensuring that  $\mathbf{r}_{i+1}$  lies within the solution space. The resulting  $s$  is the step minimizing the cost function in the gradient direction.

Figure 2.18 plots the perceptual MSE of this approach with  $\ell_2$  minimization in comparison with other algorithms. The results are averaged over the 32 images and at the same  $q$  when applicable. Figure 2.19 shows the same comparison for LabPSNR, defined in Section 2.3.2. The LED and LC values have been quantized to 8 and 10 bits, since real displays can only handle values with a limited bit depth. The effect of quantization is very small on MSE and is omitted from Figure 2.18. The plots show that the approach based on gradient descent always has higher perceptual quality than other algorithms, at the same or lower power level.

Table 2.6 reports the MSE values from Figure 2.18 split into leakage and clipping contributions, with additional data for  $\varepsilon = 0.0002$ . Clipping values are for  $\varepsilon = 0.001$ . The  $\varepsilon$  value affects the LED values output by the proposed approach but not the other algorithms, for which clipping is the same for both  $\varepsilon$ . Clipping for the proposed approach is 2.68 E-6 (13.01%) at  $q = 0$  and 3.60 E-5 (74.01%) at  $q = 100$  if optimizing for  $\varepsilon = 0.0002$ . The dotted line in Figure 2.18 draws the clipping contribution for each  $q$  of the algorithm based on gradient descent.

These results altogether show that high power algorithms tend to render with higher fidelity but neglect leakage, which is the main cause of distortion when the backlight is intense, while low power algorithms suffer mainly from clipping. Optimizing with GD in the perceptual domain finds in all cases the best trade-off between quality and power consumption. This limits color distortion, as shown by the LabPSNR data in Figure 2.19. The same figure shows that the quantization error tends to dominate leakage and clipping error at high power levels, as revealed by the tendency of the curves to saturate, while it is less important at lower power.

In order to assess the performance of the GD approach and generally verify the validity of the backlight dimming LCD model, the visual quality of the results was tested in a subjective image quality experi-

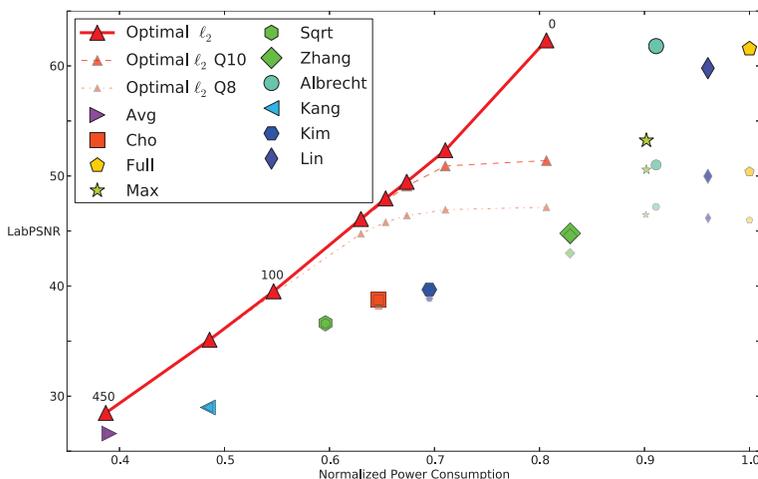


**Figure 2.18:** Distortion (MSE) vs. power trade-off of the gradient descent approach (curve) compared with other algorithms on edge-lit 16 segments backlight. Labels indicate power penalty. The dotted line plots the contribution of clipping to MSE for optimal  $\ell^2$ . References are: Max and Sqrt [25], Cho [26], Zhang [27], Albrecht [36], Kang [31], Kim [29], Lin [35].

ment. The backlight of the edge-lit display was simulated and shown on the direct-lit SIM2 screen. The borders of the screen, about 30 pixels on each side, were covered to focus the attention of the subjects to the center of the display.

Sixteen test subjects (12 men and 4 women of age ranging from 22 to 30) performed the test, all ignoring its goal and not experts in backlight dimming. The subjects sit in front of the display at a distance equal to three times its height (58.53 cm). The viewing point was perpendicular with the screen. The subjects were asked to express a preference between two versions of the same image displayed with two different backlight dimming algorithms. They were allowed to freely switch between the two images before making their choice. To limit the duration of the experiments, a set of seven images was used, three among those used in the objective evaluation (Stars image from Figure B.1, Beach (k12) and Parrot (k23) from the Kodak dataset B.3), and other four images shown in Figure B.5. All figures are in Appendix B.

The preferences of the subjects were transformed into a subjective rank order, which was then used to study the correlation between the



**Figure 2.19:** Quality (LabPSNR) vs. power trade-off of the gradient descent approach (curve) compared with other algorithms on edge-lit 16 segments backlight. Labels indicate power penalty. The smaller markers indicate quantized results for 10 bits (middle size) and 8 bits (smallest); power levels and symbols are the same as the non-quantized result. References are: Max, Avg and Sqrt [25], Cho [26], Zhang [27], Albrecht [36], Kang [31], Kim [29], Lin [35].

objective rank order based average LabPSNR and the subjective preference. The results are summarized in Table 2.7. The data show a fairly good match between subjective and objective results in terms of Spearman rank order correlation coefficient (SROCC), except for one outlier case (Stars). The average SROCC is 0.80. Gradient descent with low power weight ( $q=0$  or  $q=1$ ) outperforms all the other algorithms in terms of subjective preference in all cases except one (Volcano), where Albrecht [36] is preferred (however with significantly higher power consumption).

Both subjective and objective results show the GD algorithm outperforming all the other algorithms with the same or lower power consumption. Figure 2.20 shows two pictures of the displayed results of the Albrecht and GD algorithms for the Stars image. For the gradient descent, leakage is less annoying.

### Block-based Gradient Descent

This section presents a technique to accelerate the GD based backlight dimming algorithm. The proposal exploits the fact that the backlight

	$\varepsilon = 0.0002$		$\varepsilon = 0.001$		
	Leakage	%	Leakage	%	Clipping
Avg	2.70 E-6	0.14	1.32 E-5	0.66	1.97 E-3
Cho	1.66 E-5	11.26	7.54 E-5	36.57	1.31 E-4
Full	3.51 E-5	100	1.60 E-4	100	0
Max	2.91 E-5	98.88	1.33 E-4	99.75	3.31 E-7
Sqrt	7.33 E-6	2.15	3.46 E-5	9.38	3.34 E-4
Zhang	2.88 E-5	77.86	1.31 E-4	94.11	8.18 E-6
Albrecht	3.36 E-5	100	1.52 E-4	100	0
Kang	1.66 E-5	1.99	7.46 E-5	8.35	8.19 E-4
Kim	1.97 E-5	14.93	8.93 E-5	44.26	1.12 E-4
Lin	3.41 E-5	97.38	1.55 E-4	99.41	9.18 E-7
Opt q=0	1.79 E-5	86.99	6.03 E-5	98.48	1.28 E-5
Opt q=100	1.21 E-5	25.09	4.57 E-5	45.09	5.56 E-5

**Table 2.6:** Contribution of leakage and clipping to MSE (edge-lit).



**Figure 2.20:** Displayed results of the Stars image (as in Table 2.7) for backlight algorithm *Albrecht* [36] (left) and gradient descent with no power penalty (*GDq0*, right) [10].

has a smooth diffusion, in comparison to the pixel resolution.

GD search can improve an existing backlight by reducing the cost defined by function  $f$  (Eq. 2.46). The cost is always dependent on the error measured at pixel level (Problem 2.35) but can include also a term dependent on power consumption (Problem 2.39). Focusing initially on the image quality term, the physical error at pixel  $i$  is defined as

$$E_i = y_i - x_i, \quad (2.47)$$

<b>Exotic</b> (SROCC 0.96)			<b>Lizard</b> (SROCC 0.89)		
<i>GDq0</i>	45.17	0.948	<i>GDq1</i>	46.46	0.870
Albrecht	45.78	0.985	Albrecht	47.35	0.981
Zhang	38.01	0.932	Zhang	36.28	0.904
<i>GDq200</i>	33.52	0.763	<i>GDq400</i>	37.06	0.644
Kang	29.32	0.752	Cho	25.35	0.684
<i>GDq400</i>	23.86	0.578	Kang	24.57	0.658
Cho	22.76	0.635	<b>Beach</b> (SROCC 0.89)		
<b>Parrot</b> (SROCC 0.96)			<i>GDq1</i>	51.91	0.730
<i>GDq0</i>	51.42	0.813	Albrecht	52.92	0.855
Albrecht	51.74	0.840	Cho	39.11	0.767
Zhang	50.00	0.811	Zhang	45.25	0.800
<i>GDq100</i>	44.92	0.648	<i>GDq800</i>	23.17	0.483
Cho	38.50	0.660	Kang	20.42	0.448
<i>GDq600</i>	25.37	0.440	<b>Diver</b> (SROCC 0.70)		
Kang	25.02	0.447	<i>GDq0</i>	46.90	0.702
<b>Volcano</b> (SROCC 0.83)			<i>GDq400</i>	30.12	0.546
Albrecht	38.98	0.660	Albrecht	44.11	0.825
<i>GDq0</i>	38.19	0.513	Zhang	32.54	0.740
Zhang	38.04	0.767	Cho	23.66	0.560
Cho	25.48	0.439	<b>Stars</b> (SROCC 0.40)		
Kang	22.62	0.388	<i>GDq0</i>	39.29	0.304
<i>GDq400</i>	26.39	0.324	Albrecht	36.89	0.982
			Kang	38.58	0.504
			Cho	38.34	0.499

**Table 2.7:** Performance comparison of backlight dimming algorithms. Algorithms are listed in subjective preference order for each image; *GDqX* refers to the gradient descent algorithm with power penalty  $q=X$ ; central columns are LabPSNR, right columns are normalized power consumption [10].

where  $y_i$  is the target luminance for pixel  $i$  and  $x_i$  is the actual luminance output on the display by pixel  $i$ . Both variables are normalized between 0 and 1. If the error is measured in the perceptual domain, Eq. 2.47 becomes

$$E_i = g(y_i) - g(x_i), \quad (2.48)$$

where  $g$  (Eq. 2.25) simulates perception of luminance by the HVS. The

actual output  $x_i$  is defined by

$$x_i = b_i \times f_C(y_i, b_i), \quad (2.49)$$

where  $f_C(y_i, b_i)$  is the pixel compensation function. If hard clipping is used, then  $f_C(y_i, b_i)$  is equal to the observed compensated transmittance  $t_{C_o}$  from Eq. 2.24:

$$x_i = b_i \times \left[ \frac{l_y}{b} \right]_{\perp \varepsilon}^{\top 1}, \quad (2.50)$$

where the values of  $t_{C_o}$  are between  $\varepsilon$  and 1. Combining Eqs. 2.47-2.50, the pixel error  $E_i$  is generally expressed as

$$E_i = \begin{cases} y_i - b_i \times f_C(y_i, b_i) & \text{if physical} \\ g(y_i) - g(b_i \times f_C(y_i, b_i)) & \text{if perceptual} \end{cases}, \quad (2.51)$$

which becomes the following if using hard clipping and using Eq. 2.26 (Gamma) as  $g$ :

$$E_i = \begin{cases} y_i - b_i \times \left[ \frac{y_i}{b_i} \right]_{\perp \varepsilon}^{\top 1} & \text{if physical} \\ y_i^{1/\gamma} - \left( b_i \times \left[ \frac{y_i}{b_i} \right]_{\perp \varepsilon}^{\top 1} \right)^{1/\gamma} & \text{if perceptual} \end{cases}. \quad (2.52)$$

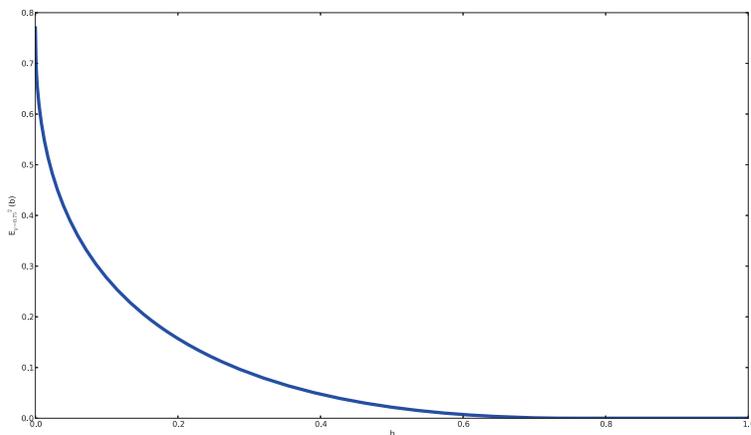
Equation 2.52 shows that the  $E_i$  is function of  $y_i$ ,  $b_i$  and of  $\varepsilon$ . For simplicity, we assume  $\varepsilon$  to be constant for all pixels. Pixels with equal  $y_i$  have the same error function, therefore Eq. 2.51 can be generalized as a function of the target luminance  $y$ :

$$E_y = \begin{cases} y - b \times f_C(y, b) & \text{if physical} \\ g(y) - g(b \times f_C(y, b)) & \text{if perceptual} \end{cases}, \quad (2.53)$$

and Eq. 2.52 becomes

$$E_y = \begin{cases} y - b \times \left[ \frac{y}{b} \right]_{\perp \varepsilon}^{\top 1} & \text{if physical} \\ y^{1/\gamma} - \left( b \times \left[ \frac{y}{b} \right]_{\perp \varepsilon}^{\top 1} \right)^{1/\gamma} & \text{if perceptual} \end{cases}. \quad (2.54)$$

For each target luminance  $y$ , it is possible to calculate the error as a function of  $b$  only:  $E_y(b)$ . The absolute or the squared error can easily



**Figure 2.21:** Example of perceptual error  $E_y(b)^2$  as a function of the backlight intensity  $b$  for  $\varepsilon = 0.0002$  and for target luminance  $y = 0.75$ .

be calculated as  $|E_y(b)|$  and  $E_y(b)^2$ , respectively. Figure 2.21 shows an example of  $E_y(b)^2$  with  $\varepsilon = 0.0002$  and  $y_i = 0.75$ . The values of the function can be stored in a LUT for a set of values of  $b$ , for example going from 0 to 1 with a fixed step. Missing values can be approximated with interpolation. The LUT would help to speed up calculations that involve the pixel error.

It can be assumed that the backlight is constant within a certain block  $B$  of pixels. The error in this block is given by the sum of the errors of the pixels it contains:

$$E_B(b_B) = \sum_{i \in B} E_i(b_B), \quad (2.55)$$

where  $E_B(b_B)$  is the total block error for pixel block  $B$  and  $b_B$  is the backlight intensity within the block. If the number of occurrences of each pixel value inside the image block (its histogram) is known, then Eq. 2.55 can be expressed as

$$E_B(b_B) = \sum_{y \in H} H_y E_y(b_B), \quad (2.56)$$

where  $H$  is the set of all the values included in the histogram and  $H_y$  is the number of occurrences of value  $y$  in the block.

The size of  $B$  determines how precisely  $E_B$  approximates the error within it. The larger  $B$ , the less accurate the estimation. Considering the PSF and the LED values, the block luminance  $b_B$  can be expressed by modifying Eq. 2.19:

$$b_B = \sum_{k=1}^M r_k h_{Bk}, \quad (2.57)$$

where  $h_{Bk}$  is the luminance contribution to block  $B$  from LED  $k$ . This contribution depends on the pixel-level PSF of LED  $k$  (e.g. it can be the average pixel-contribution in the block) and defines a block-level PSF.

It is finally possible to calculate the image error  $E$  as the sum of the block errors:

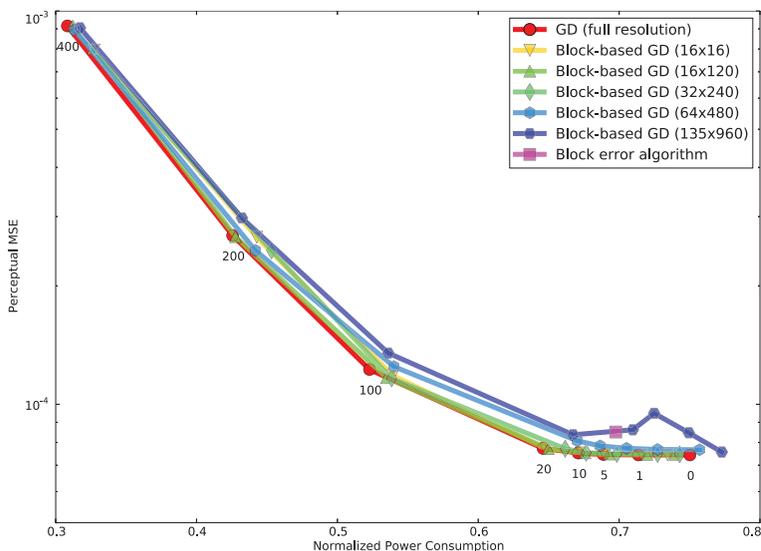
$$E = \sum_{B \in I} E_B(b_B), \quad (2.58)$$

where  $I$  is the set of blocks composing the image.

The assumption of block-wise uniform backlight allows to approximate the image error with fewer elements, as if the image were down-scaled. If the block size is  $10 \times 10$  pixels, then the number of elements to sum in Eq. 2.58 would be 100 times less than the number of all pixels. This is particularly interesting for an approach based on GD to find the optimal backlight, because it needs to calculate the image error many times to estimate the gradient and to select the step size. The values of  $E_B(b_B)$  for each block can be stored for quick access, making the GD iterations faster. The cost function for the GD approach can include a power penalty term, similarly to that of problem 2.41.

Large pixel blocks mean faster iterations and lower usage of memory (to store the block error functions), but also means lower precision. Depending on the available resources, it is possible to find an acceptable compromise between the two.

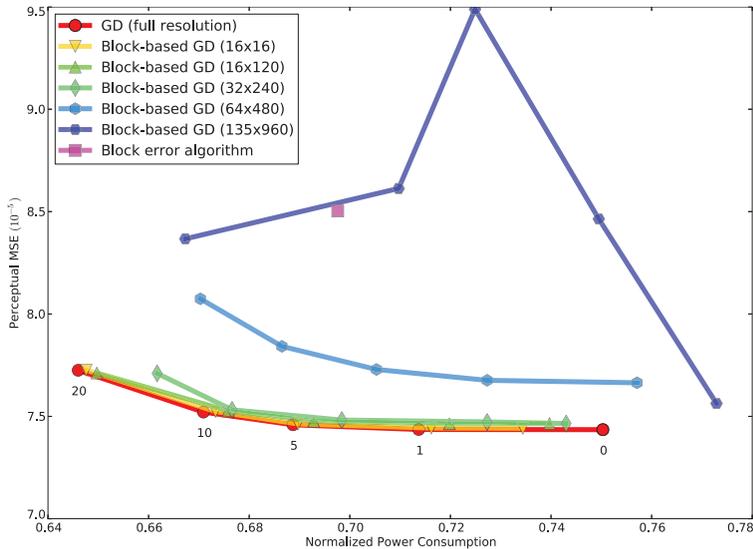
This block-based GD was implemented in Matlab and tested on the edge-lit display, having 16 segments, on the dataset shown in Figure B.1 in Appendix A. The epsilon value was set to 0.0002. The block size was set to  $16 \times 16$  pixels,  $16 \times 120$  pixels,  $32 \times 240$  pixels,  $64 \times 240$  pixels and  $135 \times 960$  (this last one matches the size of the 16 backlight segments in the screen); the maximum numbers of iterations was set to 200. The initial solution was determined as follows: for each uniform backlight block, the backlight intensity minimizing the error in the



**Figure 2.22:** Average MSE performance, over 8 images, of the block-based gradient descent with several block sizes and the full resolution gradient descent. Labels indicate power penalty.

block was found; then, for each segment, the corresponding LED value was set to the maximum of these block backlight intensities within the segment. The original GD algorithm was used for comparison, with a maximum number of iterations of 40; the starting solution is provided by the Min-Max sub-pixel approach (Section 2.3.3), minimizing the same norm at a downscaling factor of 8. Both algorithms used step search based on the golden ratio. Several power penalties  $q$  were used to calculate the optimum at different power levels. Average MSE and PSNR have been measured in the perceptual domain. Figures 2.22 and 2.24 plot the results; Figures 2.23 and 2.25 zoom to the low error, high power consumption region. The higher MSE / lower PSNR peak for the  $135 \times 960$  block sizes at power consumption about 0.73 is caused by one image (*Pedestrian*), which appears to suffer more than the others from the assumption of uniform backlight with large blocks.

For both MSE and PSNR, the block-based GD gets very close to the full resolution GD, with little decrease in quality and small variations in power consumption. For larger blocks, it deviates more from the optimum. However, results are still high quality (in the high power range

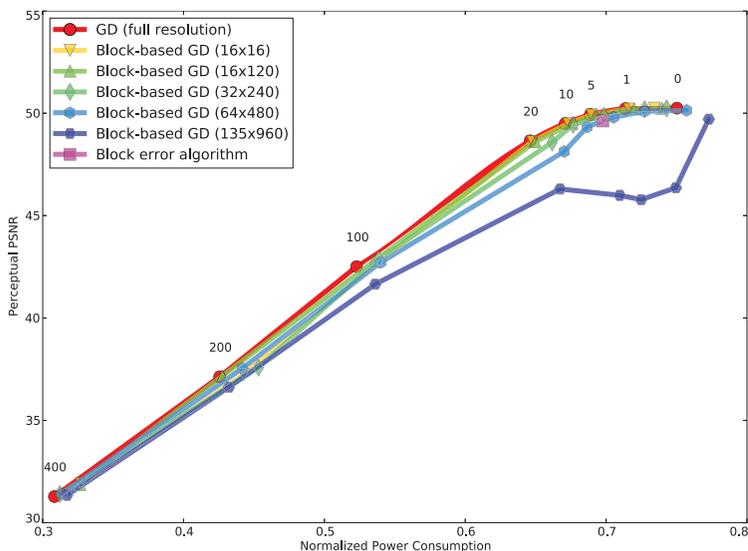


**Figure 2.23:** Average MSE performance, over 8 images, of the block-based gradient descent with several block sizes and the full resolution gradient descent (Zoom). Labels indicate power penalty.

PSNR is always above 45 dB) and the complexity is strongly reduced. Table 2.8 shows the execution times of the block-based GD for different block sizes and  $q = 0$ . It can be seen that larger blocks mean less time required to calculate all the block error functions and averagely faster iterations. The average number of iterations is also reduced, probably due to the fact that the image error is defined by less elements (the blocks of uniform backlight). These results are also dependent on the software implementation, however they give a clear indication of how the block size affects execution time.

Overall, these preliminary results show that the block-based gradient descent approach can calculate nearly optimal backlights in a shorter time, if compared to the full resolution GD. Execution time and precision decrease as the block size increases.

**Algorithm Based on Block Error** The performance of the block-based GD for 16 blocks is reasonably good, considering that it is based on the assumption that the backlight is constant within a  $135 \times 960$

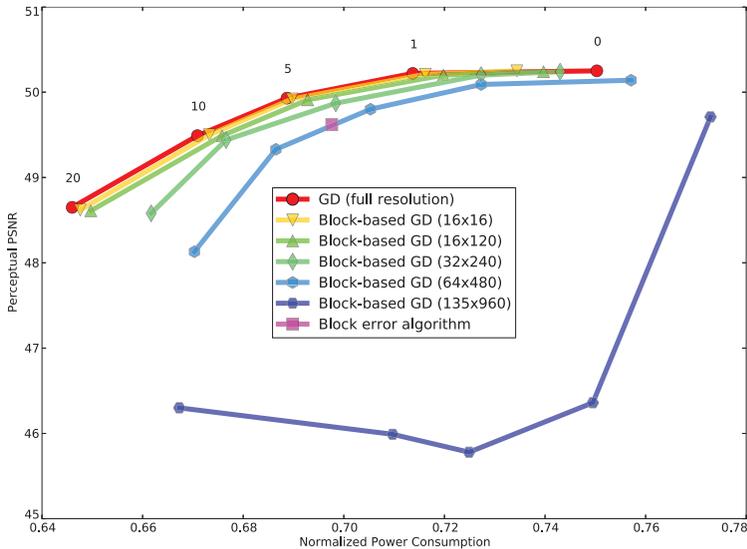


**Figure 2.24:** Average PSNR performance, over 8 images, of the block-based gradient descent with several block sizes and the full resolution gradient descent. Labels indicate power penalty.

	Time err. (s)	Avg. iter. time (s)	# Iter.
GD	-	97.85	39.38*
$16 \times 16$	2.9803	0.1248	102.50
$16 \times 120$	0.5480	0.0385	60.38
$32 \times 240$	0.3175	0.0255	60.88
$64 \times 480$	0.2342	0.0223	70.25
$135 \times 960$	0.2330	0.0219	22.25

**Table 2.8:** Comparison of execution time of the block-based GD approach for several block sizes (with  $q = 0$ ) against the regular GD, in terms of the time required to calculate the block error curves (Time err.), the average iteration time and the average number of iterations per image. \* The max number of iterations for the GD approach was set to 40, whereas for the block-based GD it was set to 200.

pixels large block, which is not accurate. If the block size coincides with the backlight segments, as in this case, it is possible to design a dimming algorithm exploiting the block error (Eq. 2.55). The algorithm simply sets the LED intensity of each segment to the backlight intensity minimizing the error of the corresponding block. The performance of



**Figure 2.25:** Average PSNR performance, over 8 images, of the block-based gradient descent with several block sizes and the full resolution gradient descent (Zoom). Labels indicate power penalty.

this approach is shown by the square markers in Figures 2.22 and 2.24. These preliminary results show that this algorithm can get rather close to the optimal MSE and PSNR curves obtained with the full GD approach, which is particularly interesting given its relatively low complexity.

## 2.4 Conclusion

Backlight dimming is a technique that has the potential to overcome two flaws on LCD technology, that is low contrast ratio due to leakage and power waste intrinsically caused by its functioning. However, artifacts like clipping, haloes or flickering are introduced and image quality and energy efficiency are generally in conflict. It is not a simple problem to determine what light diffusion can yield an energy efficient, high contrast and consistently pleasant image rendering. Many algorithms have attempted that, with different aims and outcomes.

This chapter presented a model of backlight dimming systems. Based on it, some algorithms with the goal of the finding optimal trade-off

---

between leakage and clipping, and penalizing power consumption to a certain degree, were introduced. The resulting backlights can achieve better results in both objective and subjective comparison with other approaches. In particular, the method based on gradient descent search can find the perceptual optimum and is highly flexible. It was also shown that it is possible to obtain nearly optimal results even with a considerable complexity reduction.

## Chapter 3

# LCD Backlight Scanning for 3D Crosstalk Reduction

This Chapter presents a model of backlight scanning for time-sequential stereo visualization on Liquid Crystal Display (LCD) and an algorithm to minimize crosstalk using the backlight. The Chapter is based on [9].

The previous Chapter presented LCD technology and the advantages that can be obtained using local dimming to reduce power consumption and increase the contrast ratio. Local backlight dimming would not be easily possible without local Light Emitting Diode (LED) backlights. As a matter of fact, local LED backlights can be used to improve image quality in many ways.

While LCDs are more often used to display Two Dimensional (2D) content, they can also be used to display stereo Three Dimensional (3D) content [48]. The illusion of stereo vision is obtained by showing two different views of the same scene to the left and the right eye; this is possible with either passive or active glasses. In the first case, left and right frame are displayed simultaneously but on different partitions of the screen (e.g. even and odd lines); the partitions emit light with polarizations perpendicular to each other and the use of glasses with properly polarized filters allows the right frame to reach only the right eye and the left frame to reach only the left eye. In the second case, left and right frames alternate on the screen sequentially; the frames are delivered to the intended eye using active shutter glasses based on Liquid Crystals (LCs), which become transparent to let only the intended frame

through and opaque to block the other. To maintain the input frame rate, the refresh rate of the display must be double (e.g. a 60Hz video requires a 120Hz refresh rate). This second method is also called *time-sequential stereoscopic visualization*.

When experiencing 3D video content, many aspects impact on the visual quality. One of these is the presence and intensity of *crosstalk*. Crosstalk is the incomplete isolation of the left and right frames so that one leaks into the other; the term is often confused with *ghosting*, which is the perception of crosstalk [49]. In time-sequential stereoscopic visualization on LCDs, crosstalk is caused by several factors, including the quality of the shutter glasses, their synchronization with the display, the viewing angle, the response time of the display pixels and the image update method of the screen [48]. The last two aspects are display dependent. In LCDs, frames are updated with a progressive scanning process where pixels are addressed in sequence, e.g. starting from the top-left pixel then proceeding left-to-right and top-to-bottom. After being addressed, each pixel completes the transition to its new state and becomes stable after a finite *response time*. This implies that, during the update process, there is more than one frame shown on the display. Depending on the characteristics of the display, this might happen constantly. This is one of the causes of crosstalk due to the display, independent from other factors like shutter glasses.

Crosstalk in time-sequential stereoscopic 3D LCD visualization can be reduced using local LED backlights, through the techniques of *backlight strobing* (also called blinking [50]) and *backlight scanning*. In backlight strobing the whole backlight is turned on only when all the pixels are stable, otherwise is turned off. Each frame is therefore shown only when all its pixels have been completely updated, assuring that no residue of the previous one is left. Unfortunately this might not be possible if the update process is too long. Conversely, backlight scanning is done during the update process. The backlight segments are turned on and off following the LC scanning to highlight the stable pixels against the others. Backlight strobing and scanning can be combined, by doing the first while the pixels are updated and the second while they are all stable. It is important to highlight that display luminance is a scarce resource when dealing with 3D LCD Television (TV) with passive glasses, because the LCs in the glasses always absorb part of the light

that goes through them, even at maximum transmittance. Both strobing and scanning reduce the emission of light, so it is very important to maximize it at acceptable crosstalk levels.

Backlight strobing is overall a simple problem that only requires to know when all pixels are stable. Backlight scanning, however, is more complex. A very simple way to scan the backlight is to turn on only one segment at a time, in sequence, following the pixel scanning. This approach has been used in [51] and in this work it is referred to as “basic backlight scanning”. Although simple, basic backlight scanning ignores display characteristics and can reduce luminance dramatically if the number of backlight segments is large. This chapter proposes a model for backlight scanning used to minimize crosstalk at a given luminance level. The model shares some aspect with the backlight dimming model of Chapter 2, like light diffusion in the backlight, and includes new aspects like pixel response time. The model is used to determine the variables and constraints of an optimization problem having a metric for crosstalk as cost function to be minimized. The constraints include conditions to ensure visual uniformity of the backlight and to set the luminance level.

## 3.1 Modeling

This section presents a model for backlight scanning including aspects such as timing, light diffusion, pixel transitions and luminance. The model is defined image-independent to avoid difficulties connected to variable luminance levels.

### 3.1.1 Timing of Scanning and Strobing

We consider the case of LCD screens which start drawing a new frame from the top-left pixel, going left-to-right and top-to-bottom. The *frame time*  $t_f$  is defined as the time between the start of two consecutive frames. On a display with a refresh rate of 120Hz,  $t_f = \frac{1}{120}$ s. One after the other, the pixels are addressed at time  $t_i$ ; in other words,  $t_i$  is the time when the pixel transition starts. The time required to address all pixels is called *addressing time* ( $t_a$ ). After being addressed, each pixel completes the transition after the *response time*  $t_r$  has passed.

The *scanning time*  $t_s$  is defined as the time during  $t_f$  when backlight scanning is performed:

$$t_s = \begin{cases} t_a + t_r & \text{if } t_a + t_r < t_f \\ t_f & \text{if } t_a + t_r \geq t_f \end{cases}. \quad (3.1)$$

By definition,  $t_s$  cannot be longer than  $t_f$ . When all the pixels are stable, there is no need to scan the backlight; strobing can be done instead by turning all segments on. Strobing lasts for a  $t_f - t_s$  long time. Therefore, it is not performed if  $t_s = t_f$ , because there is never an instant when all pixels are stable. Figure 3.1 includes a visualization of these timing concepts. In general, during  $t_s$  at least two frames are partially shown on the screen; the frames can be three or more, depending on  $t_a$  and  $t_r$ . In this work it is assumed that  $t_a < t_f$ , which guarantees that the frames are at most two, and that  $t_r < t_f$ , which ensures that each pixel becomes stable before being addressed again.

During one loop of backlight scanning lasting  $t_s$ , the backlight is updated  $Q$  times at regular intervals. The updates happen at time  $t_j$ , where  $j = 0, \dots, Q - 1$ , so that

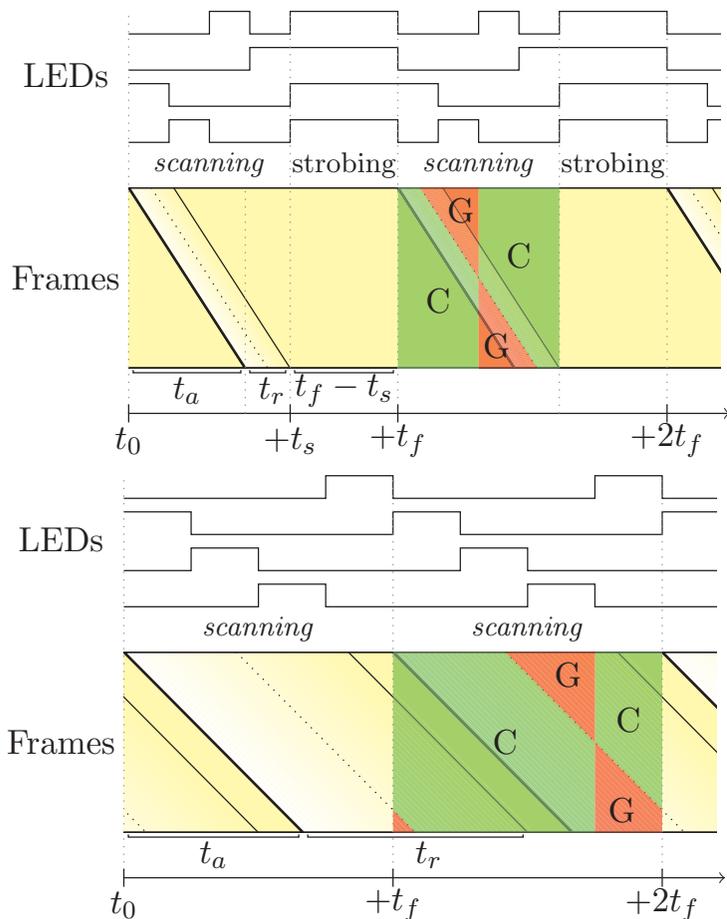
$$t_{j+1} = t_j + \frac{t_s}{Q}. \quad (3.2)$$

During the time interval  $[t_j, t_{j+1})$ , or equivalently during  $[t_j, t_j + \frac{t_s}{Q})$ , the backlight does not change. For simplicity,  $t_0$  is assumed to coincide with the beginning of the new frame, that is when the top-left pixel is addressed.

### 3.1.2 Backlight

Light diffusion in local LCD backlights has been discussed in Section 2.3.2. For backlight scanning, the modeling must be extended to include the fact that the signals  $r_k$  controlling the duty cycles of the LEDs in the backlight vary in time. The backlight intensity  $b_i(t)$  at pixel  $i$  at time  $t$  is defined as follows, based on Eq. 2.19:

$$b_i(t) = \sum_{k=1}^M h_{ik} r_k(t), \quad (3.3)$$



**Figure 3.1:** Backlight scanning/strobing timeline for  $t_s < t_f$  with short  $t_a$  and  $t_r$  (top) and for  $t_s = t_f$  with larger  $t_a$  and  $t_r$  (bottom); in the latter case strobing does not occur; the vertical axis of Frames represent vertical screen position; the shading indicates pixel transition to the new frame; the red and green overlays show when the frames are considered correct (C) or ghost (G); the square waves are the LED control signals.

where  $h_{ik}$  is the contribution of LED  $k$  to pixel  $i$ ;  $r_k(t)$  and  $b_i(t)$  are valued between 0 and 1. Uniform backlight is assumed, therefore all  $h_{ij}$  for pixel  $i$  sum up to 1 and consequently all  $h_{ij}$  sum up to the number of pixels, defined as  $N$ .

During  $[t_j, t_j + \frac{t_s}{Q})$ , the LED values  $r_k(t)$  do not change. Consequently,  $b_i(t)$  does not change either. The backlight luminance for pixel  $i$  and the value of LED  $k$  during this time are indicated as  $r_{kj}$  and  $b_{ij}$ , respectively:

$$b_{ij} = \sum_{j=1}^M h_{ik} \cdot r_{kj}. \quad (3.4)$$

### 3.1.3 Frames

To model crosstalk, the definitions of *correct frame* and *ghost frame* (so called because it causes ghosting) are introduced. The correct frame is the frame that is being shown on more than  $\frac{N}{2}$  pixels; the ghost frame is the frame shown on the remaining pixels.

When two frames are displayed on the screen, previous/current frame and correct/ghost frame have a relation. The current frame starts to be updated at  $t_0$  and, at that time, it is the ghost frame; clearly, the previous frame is the correct frame. When half of the pixels have changed to the current frame, the previous frames switches from correct to ghost, while conversely the current frame becomes the correct one. The rigorous definition of the moment when the switch happens depends on the pixel transition function (modeled in the next Section). For simplicity, this work defines this instant to be  $t_0 + \frac{t_a + t_r}{2}$ . Figure 3.1 includes an illustration of these concepts.

### 3.1.4 Pixel Transition and State

In a LCD, the luminance emitted by pixel  $i$  is proportional to the intensity of the backlight and to the pixel transmittance. To keep the model image-independent, transmittance is ignored. However, it is necessary to know the state of the pixel in its transition from one frame to the other, because backlight scanning and strobing are defined depending on this state. This transition is expressed with function  $f(t)$ . This function is characterized by the response time  $t_r$ , which depends on the LC characteristics and on the initial and final gray values. The model considers only one response time representative of all the gray-to-gray transitions, to keep the model image-independent. This time could be, for instance, the largest response time or another time which approximates the re-

sponse time of a significant selection of gray-to-gray transitions. The function  $f(t)$  is real, valued between 0 and 1 and is increasing in  $[0, t_r]$ ; its value is 0 if  $t < 0$  and 1 if  $t > t_r$ . The LC transitions have been studied and modeled in many previous works [52]–[54]. For simplicity, we assume a linear transition from 0 to 1 in  $[0, t_r]$ .

In relation to the correct and ghost frame, the *state* of pixel  $i$  can be expressed with the periodical function  $s_i(t)$ , which is based on  $f(t)$  and has real values between 0 and 1. When  $s_i(t)$  is 1, the pixel is in the correct frame; when  $s_i(t)$  is 0, it is part of the ghost frame; intermediate values mean that the pixel is transitioning from one frame to the other. The function  $s_i(t)$  is periodical with period  $t_f$ , because the pixels are refreshed over the same period. Its complete definition of is

$$s_i(t) = \begin{cases} 1 - f((t - t_i) \bmod t_f) & \text{if } t \in [t_i, \frac{t_a + t_r}{2}] \\ & \text{or if } t \in [\frac{t_a + t_r}{2}, t_i] \\ f((t - t_i) \bmod t_f) & \text{otherwise} \end{cases} \quad (3.5)$$

During the time interval  $[t_i, \frac{t_a + t_r}{2}]$  (or  $[\frac{t_a + t_r}{2}, t_i]$  if  $t_i > \frac{t_a + t_r}{2}$ ), the value of  $f(t)$  is inverted to model the fact that the transition is from the correct frame to the ghost frame. Figure 3.2 shows examples of  $s_i(t)$ .

The state function can be averaged over the time interval  $[t_j, t_j + \frac{t_s}{Q}]$  to define  $s_{ij}$ , the average state of pixel  $i$  during this time interval:

$$s_{ik} = \frac{Q}{t_s} \int_{t_k}^{t_k + \frac{t_s}{Q}} s_i(t) dt. \quad (3.6)$$

Figure 3.3 illustrates the concept of average pixel state.

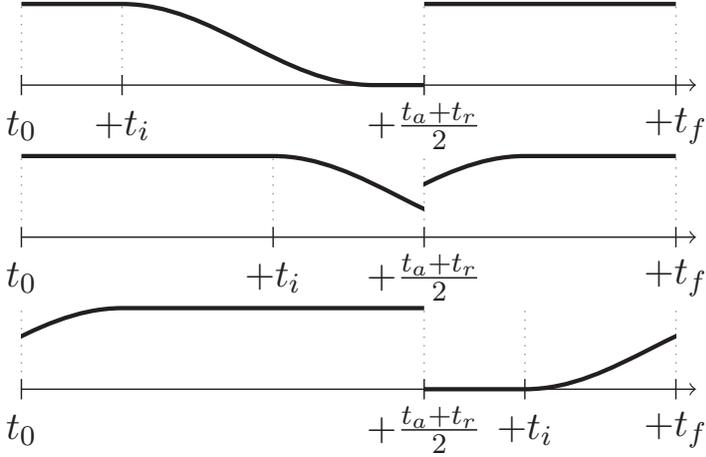
### 3.1.5 Luminance

The contribution of pixel  $i$  to the luminance of the correct frame and the ghost frame are defined as  $l_{C,i}(t)$  and  $l_{G,i}(t)$ , respectively:

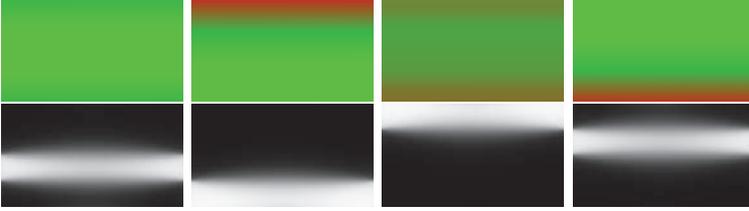
$$l_{C,i}(t) = s_i(t)b_i(t), \quad (3.7)$$

$$l_{G,i}(t) = (1 - s_i(t))b_i(t). \quad (3.8)$$

Equation 3.8 reflects the fact that the light not going through the correct frame must go through the ghost frame. The average  $l_{C,i}$  during  $[t_0, t_0 +$



**Figure 3.2:** Three examples of pixel state transitions  $s_i(t)$  with different  $t_i$  (pixel addressing time).



**Figure 3.3:** Example of average pixel state (top) and backlight diffusion (bottom) during scanning time with 4 segments and 4 updates; green color corresponds to an average state mostly in the correct frame, while red corresponds to the ghost frame.

$t_s$ ) can be calculated by integration:

$$l_{C,i} = \frac{1}{t_s} \int_{t_0}^{t_0+t_s} s_i(t) b_i(t) dt. \quad (3.9)$$

During  $[t_0, t_0 + t_s)$ , the backlight is updated  $Q$  times and it is constant during  $[t_j, t_j + \frac{t_s}{Q})$  with  $j = 0, 1, \dots, Q - 1$ . Combining Eq. 3.9 with Eqs. 3.4 and 3.6 the result is

$$l_{C,i} = \frac{1}{Q} \sum_{j=0}^{Q-1} b_{ij} s_{ij}. \quad (3.10)$$

Assuming a uniform backlight, the luminance  $L_C$  of the correct frame can be expressed as the average of all the pixel luminances:

$$L_C = \frac{1}{N} \sum_{i=1}^N l_{C,i}. \quad (3.11)$$

Combining Eq. 3.11 with Eqs. 3.10 and 3.4 gives

$$L_C = \frac{1}{NQ} \sum_{i=1}^N \sum_{j=0}^{Q-1} \sum_{k=1}^M h_{ik} s_{ij} r_{kj}. \quad (3.12)$$

Equation 3.12 can be rewritten as

$$L_C = \frac{1}{NQ} \sum_{k=1}^M \sum_{j=0}^{Q-1} g_{kj} r_{kj}. \quad (3.13)$$

where  $g_{kj} = \sum_{i=1}^N h_{ik} s_{ij}$ . As clear from Eq. 3.13,  $L_C$  is a linear combination of the LED values  $r_{kj}$  with the values  $g_{kj}$  as coefficients.

The luminance of the ghost frame  $L_G$  can be calculated in the same way as  $L_C$ , but using  $l_{G,i}$  instead of  $l_{C,i}$ . It can be shown that

$$L_G = \frac{1}{NQ} \sum_{i=1}^N \sum_{k=0}^{Q-1} b_{ik} - L_C, \quad (3.14)$$

which shows again that the backlight flows through either the correct or the ghost frame. From the central term of Eq. 3.14 the frame luminance  $L$  can be defined:

$$L = \frac{1}{NQ} \sum_{i=1}^N \sum_{k=0}^{Q-1} b_{ik}, \quad (3.15)$$

from which follows that

$$L = L_G + L_C. \quad (3.16)$$

## 3.2 Optimal Backlight Scanning

The model that has just been presented has been used to formulate an optimization problem, the goal of which is to minimize crosstalk at a given luminance level using backlight scanning. The problem includes a cost function for crosstalk and a constraint to assure light uniformity.

### 3.2.1 Crosstalk

In [51], the authors measure crosstalk with a metric using black and white frames and following international standards (Section 17.1, Eq. 27 of [55]). This metric, however, is designed for measurements with a light-meter and takes in to account the impact of shutter glasses. To measure crosstalk as a function of the display only and to avoid image dependency, the following metric is used instead:

$$Crosstalk = \frac{L_G}{L_C}, \quad (3.17)$$

which combined with Eq. 3.16 gives

$$Crosstalk = \frac{L}{L_C} - 1. \quad (3.18)$$

Minimizing Eq. 3.18 is equivalent to minimizing Eq. 3.17 which, for fixed  $L$ , is equivalent to maximizing  $L_C$ .

### 3.2.2 Constraints

It is necessary to enforce a constraint on  $r_{kj}$  ensuring a uniform backlight. It is assumed that the  $Q$  updates of the backlight during  $t_s$  are fast enough to avoid visible flickering and to produce a sensation of constant backlight. It is then required that the average value of LED  $j$  during the time interval  $[t_0, t_s)$  is constant and equal for all LEDs:

$$\frac{1}{Q} \sum_{j=0}^{Q-1} r_{kj} = d \quad k = 1, \dots, M. \quad (3.19)$$

The constant  $d$  is valued within 0 and 1. Remembering that the sum of all the  $h_{ik}$  elements is  $N$  (see Section 3.1.2) and combining Eqs. 3.4, 3.15 and 3.19, the result is

$$L = d, \quad (3.20)$$

which means that the frame luminance is equivalent to the average LED value during scanning time.

### 3.2.3 The Optimization Problem

The goal of optimization is the minimization of crosstalk. This is equivalent to maximizing  $L_C$ , if  $d$  is fixed. The variables of the problem are the  $M \times Q$  LED values  $r_{kj}$ . The constraints are the uniformity constraint (Eq. 3.19) and the range of valid values of  $r_{kj}$ . The problem is thus:

$$\begin{aligned} & \text{maximize} && \sum_{k=1}^M \sum_{j=0}^{Q-1} g_{kj} r_{kj} \\ & \text{subject to} && \sum_{j=0}^{Q-1} r_{kj} = Qd \quad k = 1, \dots, M \\ & && 0 \leq r_{kj} \leq 1 \end{aligned} \tag{3.21}$$

Realistic values of  $M$  are in the order of tens, while  $Q$  should not go beyond the thousands. The number of variables  $r_{kj}$  is therefore in the order of tens of thousands. The value function to maximize is a linear combination of these variables. Overall, the complexity of the problem can be considered low. It might be more computationally demanding to calculate the  $g_{jk}$  coefficients, depending on  $f(t)$ . Nevertheless, the solution to the problem (for all the luminance levels of interest) needs to be solved only once per display. After the optimal scanning control signals are known, they can be used to control the scanning backlight of the LCD without any additional cost.

## 3.3 Experiments and Results

The optimization based optimal backlight scanning has been tested and compared against basic backlight scanning, where the segments are turned on at full power ( $r_{kj} = 1$ ) in sequence and only one at a time. This means that the constraint given by Eq. 3.19 is respected and that  $d = \frac{1}{M}$ , which is also the frame luminance  $L$ .

The Full High Definition (HD) edge-lit display with 16 segments described in Section 2.3.2 was used for the simulations. The segments, placed in 8 rows and 2 columns, were grouped to emulate coarser backlights (e.g. 4 rows and 1 column). We considered the cases of 2, 4 and 8 rows and 1 column and additionally that of global backlight where all  $r_{ij}$  are set to  $d$ .

The optimization approach was implemented in Matlab using the CVX package. The frame time  $t_f$  was set to  $\frac{1}{120}$ s = 8.33ms. The addressing time  $t_a$  was set to  $0.75 \times t_f$ . Three response times  $t_r$  were considered: 0ms (ideal, instantaneous response), 2ms and 4ms. The pixel response  $f(t)$  was modeled linearly, for simplicity. The number of backlight updates  $Q$  was set equal to the number of pixel lines (1080); this is justified by the very fast response time of LEDs, which is in the order of few microseconds, or less. The minimum crosstalk was calculated for different values of  $d$  ranging from 0 to 1 with a step of 0.01.

Figure 3.4 shows the results for  $t_r = 0$ ms,  $t_r = 2$ ms and  $t_r = 4$ ms. The plots account for backlight scrolling and strobing combined. While strobing, the backlight is fully turned on and all pixels are stable, therefore the frame luminance is 1 and the crosstalk is 0. The luminance and crosstalk measures combined for scrolling and strobing are obtained from a weighted average where the weight are  $\frac{t_s}{t_f}$  for scrolling and  $\frac{t_f - t_s}{t_f}$  for strobing. Combined luminance over  $t_f$  is therefore  $\frac{t_s}{t_f} L_C + \frac{t_f - t_s}{t_f}$ , while average crosstalk is equal to  $\frac{t_s}{t_f} \frac{L_G}{L_C}$ .

It can be seen that optimal backlight scanning always improves the tradeoff between luminance and crosstalk compared to basic scanning. For instance, with  $t_r = 4$ ms and 2 segments, basic scanning has 50% luminance and  $1.335 \times 10^{-1}$  crosstalk. Optimal scanning can either reduce crosstalk to  $2.895 \times 10^{-2}$  at the same luminance level or increase luminance to 83% with the same crosstalk. The relative improvement is even larger for 4 and 8 segments. At the same luminance level, having more backlight segments allows to reduce crosstalk further. This is expected as, in these experiments, segments are grouped together to simulate coarser backlights, which limits the degrees of freedom of the optimization. It can also be seen that the size of the improvement varies with the luminance level, and that the improvement from 2 to 4 segments is rather large if compared to that from 4 to 8 segments. This suggests that 4 backlight segments could be “good enough” to obtain acceptable results.

Figure 3.5 shows how the backlight control signals have changed after optimization in comparison to basic scanning, for the  $t_r = 4$ ms case. The figure suggests that the crosstalk reduction is achieved by turning on more LEDs simultaneously and by concentrating the light output

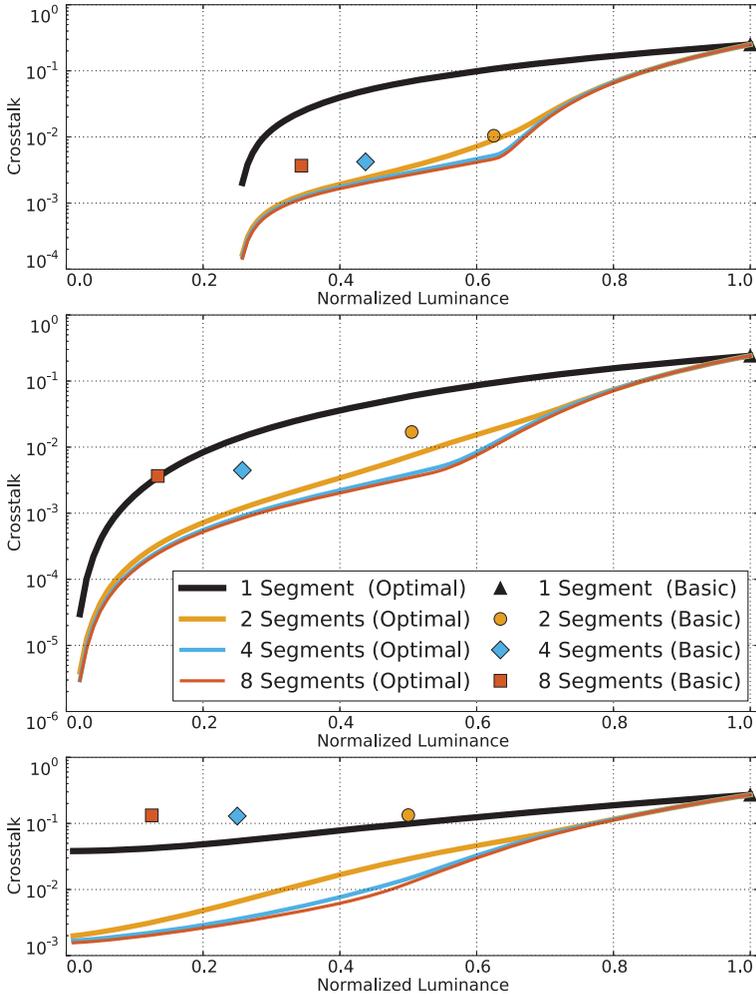
when it is most convenient to increase  $L_C$ . This causes the backlight to be turned off for part of the scanning time, creating a sort of complementary strobing that turns the backlight off when doing otherwise would increase crosstalk. The figure also shows that the shape of the square waves controlling the LEDs is the same. This indicates that optimal backlight scanning is obtained by properly “shifting” the waves to the most convenient time intervals.

These results, obtained from simulations, are illustrative of the improvements that can be obtained by optimizing backlight scanning, even if some simplifications have been used (i.e. linear transition function). In any case, the model is general enough to allow more precise simulations.

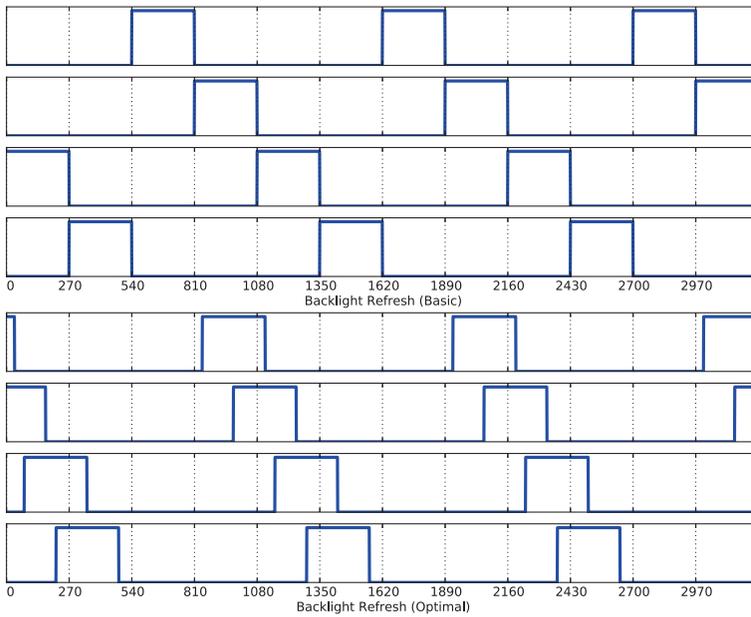
As a final note, the same experiments have been run on the same screen but downscaled by a factor of 10, both at backlight and at pixel level. The results were nearly identical, showing that it is acceptable to reduce the complexity of the problem by downsampling without significant loss of precision.

### 3.4 Conclusion

LCDs, when used to display 3D content using time-sequential visualization with active shutter glasses, can have high stereo crosstalk and consequently provide a poor perceptual performance. However, dynamic backlight can limit crosstalk with the technique of backlight scanning. This chapter presented a model for backlight scanning which was used to formulate an optimization problem, where the cost function to minimize is a measure of crosstalk and the constraints include the luminance level. Simulation results show that the best tradeoffs between crosstalk and luminance can be obtained by concentrating the emission of light when this can reduce crosstalk. It was also shown that more backlight segments yield better results, however 4 segments appear to be enough for acceptable performance, as the improvement associated to additional segments is decreasing. The optimization problem has a relatively low complexity, because the cost function is linear and the number of variables is in the range of tens of thousands. Moreover, the optimal scanning signals controlling the backlight need to be determined only once per screen, and downsampling the display in the model does not affect the final result significantly.



**Figure 3.4:** Results of the experiment for response time  $t_r = 0\text{ms}$  (top),  $t_r = 2\text{ms}$  (middle) and  $t_r = 4\text{ms}$  (bottom). The lines indicate the optimal crosstalk at different luminance levels for a different number of backlight segments. The markers indicate the performance of basic backlight scanning. The measures account for both phases of backlight scanning and backlight strobing.



**Figure 3.5:** LED control signals before (top, basic backlight scanning) and after optimization (bottom) with 4 backlight segments.

## Chapter 4

# Conclusion and Future Work

The presence of images and video has pervasively spread in modern society. Huge amounts of information, particularly for entertainment purposes, are coded, transmitted and displayed continuously. Visual quality clearly plays a major role in this context, particularly for Television (TV) screens. Modern Liquid Crystal Displays (LCDs) allow to dynamically dim the backlight. This can improve the typically low contrast ratio of this display technology and generally improve image quality. Dimming the backlight also reduces power consumption, which is particularly important given the growing attention on energy efficiency and sustainable development.

This thesis presented several techniques to improve image quality through the use of dynamic local backlights or through post-processing. A model of backlight dimming systems, including aspects as light diffusion and light perception, was established and used to develop advanced dimming algorithms. It was shown that it is possible to determine the optimal backlight for an image, given the characteristics of the display and a cost function. By penalizing power consumption, it was possible to draw the curve of optimal tradeoffs between image quality and energy efficiency. This was done thanks to the backlight dimming algorithm based on gradient descent, presented in this work, which outperforms competing algorithms in both objective and subjective assessment. Efforts have been made to reduce its high complexity and bring it closer

to a real-time implementation.

Dynamic local backlights can also be used to improve the visual quality of Three Dimensional (3D) video content is displayed using LCDs and active shutter glasses. Stemming from the model of backlight dimming systems, backlight scanning for crosstalk reduction was formulated as an optimization problem where the cost to minimize is a measure of crosstalk and the constraints include a fixed luminance level. It was shown that it is possible to preserve luminance and reduce crosstalk (or conversely increase luminance while keeping crosstalk below a threshold) by properly shifting the signals controlling the light sources in the backlight.

The main part of the Ph. D. project, local backlight dimming, can continue in future work. The gradient descent algorithm needs to be refined to be applied to video signals: the backlight of a given frame can be used as starting solution to calculate the backlight of the next frame. The assumption is that the optimal backlights of consecutive frames should be relatively close. One of the main problems with temporal backlight dimming is the annoying flickering artifact, where the global or local backlight intensity fluctuates rapidly between different luminance levels. It is necessary to design a perceptual model of flickering, in order to determine when flicker occurs in the simulated backlight system. The model can be used to develop flickering metrics or to design flicker-free algorithms. For instance, the step selection of the gradient descent approach could include a flickering metric in the criteria. Alternatively, advanced de-flickering filters can be implemented. Finally, complexity should always be considered. To bring advanced dimming algorithms on actual TV sets, it is necessary to introduce approximations that make them suitable for real-time implementations. This is again a question of making trade-offs between quality and complexity.

# Appendices

# Appendix A

## Ph. D. Publications

## A.1 Modeling LCD Displays with Local Backlight Dimming for Image Quality Assessment

J. Korhonen, **N. Burini**, S. Forchhammer, and J. M. Pedersen, “Modeling LCD displays with Local Backlight Dimming for Image Quality Assessment”, in *Proceedings of SPIE*, vol. 7866, 2011, p. 786 607

Reference: [2]

## Modeling LCD Displays with Local Backlight Dimming for Image Quality Assessment

Jari Korhonen<sup>a</sup>, Nino Burini<sup>a</sup>, Søren Forchhammer<sup>a</sup>, and Jesper M. Pedersen<sup>b</sup>  
<sup>a</sup>Dept. of Photonics Eng., Tech. Univ. of Denmark, Ørstedts Plads, 2800 Kgs. Lyngby, Denmark;  
<sup>b</sup>Bang & Olufsen, Peter Bangs vej, 7600 Struer, Denmark

### ABSTRACT

Traditionally, algorithm-based (objective) image and video quality assessment methods operate with the numerical presentation of the signal, and they do not take the characteristics of the actual output device into account. This is a reasonable approach, when quality assessment is needed for evaluating the signal quality distortion related directly to digital signal processing, such as compression. However, the physical characteristics of the display device also pose a significant impact on the overall perception. In order to facilitate image quality assessment on modern *liquid crystal displays* (LCD) using *light emitting diode* (LED) backlight with local dimming, we present the essential considerations and guidelines for modeling the characteristics of displays with *high dynamic range* (HDR) and locally adjustable backlight segments. The representation of the image generated by the model can be assessed using the traditional objective metrics, and therefore the proposed approach is useful for assessing the performance of different backlight dimming algorithms in terms of resulting quality and power consumption in a simulated environment. We have implemented the proposed model in C++ and compared the visual results produced by the model against respective images displayed on a real display with locally controlled backlight units.

**Keywords:** Liquid crystal display, LED backlight, Local backlight dimming, Image quality assessment

### 1. INTRODUCTION

In this paper, we focus on a specific class of displays, namely LCDs using locally adjustable LED backlight segments. LC displays require a backlight, since liquid crystal (LC) layer acts as a “shutter”, blocking part of the backlight in order to obtain the desired intensity. In color screens, pixels are normally formed by three sub-pixels, one for each of the RGB color components. The transmittance of each sub-pixel can be controlled by an independent signal. The backlight can be based on various technologies, like fluorescent lamps or LEDs. Recently, LEDs have become more and more popular, thanks to their longer lifetime, wider color gamut, faster response and easy control. The efficiency of LEDs has been doubling every third year since 1960s, and recently LEDs have bypassed fluorescent lamps in terms of power efficiency.

Backlight dimming is motivated primarily by two factors: energy saving and improved contrast. Global backlight dimming of a uniform backlight is the simplest solution adopted in some early LC displays. With local dimming the energy savings and contrast may be further improved. Assuming that the image shown on the display contains both dark and bright regions, significant amount of energy can be saved without loss of quality by using lower backlight level in the dark regions. This is why the original intended luminance of the dark pixels can be maintained by adjusting the LC transmittance level upwards accordingly, when backlight intensity is reduced. This procedure is referred as brightness preservation. Contrast can be improved, since some backlight is normally leaking through the LC layer in the dark regions, and the perceptual impact of light leakage can obviously be alleviated by using lower backlight level. Local backlight dimming also reduces the production of heat, consequently decreasing mechanical distortions of the panel due by high temperature [1].

In the literature, several different approaches have been proposed for backlight dimming and brightness preservation [2-4]. However, less effort has been invested on evaluating and comparing these approaches. Basically, an optimal scheme would consume as little energy as possible, while preserving the image quality as well as possible. Energy saving is easy to estimate, since energy consumption of a LED is assumed to be linearly dependent on its intensity [2]. Quality assessment is not that straightforward, since subjective image quality assessment typically requires significant workload

and resources, and the known objective (ie. algorithm based) image and video quality metrics have not been designed with backlight dimming in mind.

Different objective image and video quality metrics have been studied extensively during the past years. To name a couple, Peak signal-to-noise ratio (PSNR) is computed from the mean squared error (MSE) between the original reference signal and the assessed test signal, and it is still widely used and gives relatively meaningful comparative results with fixed content and distortion type. Structural similarity (SSIM) index is another well-known objective metric for image quality assessment. Compared to PSNR, SSIM is more complex, but SSIM index typically match better to the subjective perception than PSNR [2].

PSNR, SSIM and other similar metrics are typically computed from the luma values representing the brightness of each pixel in the reference image and test image. For example, when the raw image data is stored in YUV format, the luma component Y can be used as input signal for objective quality measurements. Sometimes chroma components (U,V) are also involved. Since the relationship between perceived brightness and luminance is not linear, luma values are usually gamma corrected to make the quantization steps represent perceptually uniform differences in the brightness levels. Gamma correction is not just useful for quality assessment, but it also allows more efficient packing of luma information without loss of perceived quality.

In this paper, we attempt to address the issue of quality assessment in the context of HDR displays employing local backlight dimming, and propose a conceptual framework for reconstructing an image from pixel values and backlight values that can be used as input for objective image quality metrics. The rest of this paper is organized as follows. In Section 2, we review the related work and open issues in local backlight dimming in more detail. In Section 3, we describe our approach for modeling a display with locally adjustable backlight units. In Section 4, we present some experimental results with a real display. In Section 5, the results and open issues are discussed, and finally, the concluding remarks are given in Section 6.

## 2. LOCAL BACKLIGHT DIMMING

The basic concept of local backlight dimming is illustrated in Fig. 1. The digital image signal is first sent to the backlight dimming module in either RGB or YUV format. Then, the backlight dimming module generates two signals: one is the traditional video signal that is controlling the transmittance of the R, G, and B subpixels in the LC layer. The second signal is steering the backlight segments, either individual LEDs, or segments consisting of a group of LEDs. Between backlight elements and the LC layer there is a diffuser plate.

In the simplest form of the concept, the backlight comprises one segment only; instead of local dimming, this would be referred as global dimming, or 0-D dimming. Multiple segments can span across the screen in different forms; segments forming columns or rows would allow 1-D dimming, and grids 2-D dimming, respectively. Some backlights can generate colored light for R, G, and B components separately and perform 3-D dimming, since color is considered as a dimension. It is also possible to classify backlights as direct and edge backlight. In direct backlight, the light sources are placed behind the LC panel, while in edge backlight they are placed on one or more sides of the panel, allowing only 1-D dimming. On the one hand, direct backlight is more efficient and allows finer segmentation compared to edge backlight, but on the other hand, edge backlight allows the construction of thinner screens.

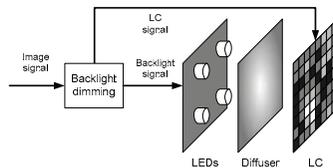


Figure 1. Concept of local backlight dimming illustrated.

**2.1 Modeling backlight diffusion**

Traditionally, the function of the diffuser plate is to spread the light from individual light sources as smoothly as possible over the whole display surface, in order to guarantee the best achievable homogeneity of luminance in different parts of the screen. However, the idea of local dimming has challenged this design target, since the differentiation in local brightness levels is in fact the goal. Finding the optimal compromise between smooth spreading of light and capability of brightness differentiation is an interesting challenge in physical design in diffuser plates in the future. Nevertheless, in this paper we still rely on assuming that diffuser plates have homogeneous light diffusion properties, i.e. the observed diffusion of light does not depend on the position of the light source behind the diffuser.

If the brightness of the backlight on the diffuser plate at each specific pixel  $(i, j)$  is known, the perceived brightness of that pixel denoted with  $I(i, j)$ , can be expressed as a product of backlight brightness  $A(i, j)$  and LCD transmittance  $T(i, j)$ , as given in Eq. 1. In color displays, the intensities are computed separately for R, G and B components, using the respective transmittance levels  $T_R, T_G$  and  $T_B$ .

$$I(i, j) = A(i, j) \cdot T(i, j) \tag{1}$$

The backlight resolution is usually significantly lower than LCD resolution, and the brightness of each pixel is typically contributed by light arriving from several LEDs, mixed on the diffuser plate. This is why a diffusion model is needed to estimate the backlight intensity at certain pixel position. A common assumption in the literature is that the luminance of a pixel on the diffuser plate is the sum of the attenuated luminance intensities of the contributing LEDs [2,3]. The attenuation factor is dependent on the distance to the LED. The basic assumption is that the point spread function of the luminance on the diffusion plate follows roughly Gaussian distribution, centered around the midpoint of the light source. Therefore, we have constructed a diffusion model given in Eq. (2), where  $n$  is the number of contributing LEDs,  $L_k$  is the luminance intensity of LED  $k$ ,  $d_k(i, j)$  is the distance between LED  $k$  and pixel  $(i, j)$ , and  $\delta$  is a parameter that is specific for the display. It is worth noting that  $n$  does not need to cover all the LEDs in the display, but only those that are close enough to contribute the luminance of the pixel in question. Fig. 2. shows a one-dimensional illustration how the overall luminance is formed from contributing luminance intensities.

$$A(i, j) = \sum_{k=1}^n L_k \exp(-\delta d_k(i, j)^2) \tag{2}$$

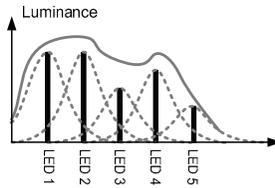


Figure 2. Diffusion model based on additive local light intensities.

Unfortunately, the practical light distribution is influenced several factors, such as the reflecting characteristics and potential seams on the background plate, optical properties of the diffuser plate and even the properties of the LEDs. This is why the basic assumption of purely Gaussian light distribution is usually not realistic. In addition, many practical

displays have local backlight segments consisting of several LEDs adjusted simultaneously, and in this case it could be reasonable to use a combined spread function. More realistic models would require accurate luminance measurements on the real display. If this is not feasible, improved models can also be developed by visual analysis of the practical light distributions and heuristic adjustment of model parameters.

## 2.2 Perceptually uniform coding

Aydın et. al. have reported that even though the commonly used transform functions for gamma correction provide perceptually uniformly scaled pixel values for darker displays, this is no longer true for brighter HDR displays [6]. The authors have defined experimentally a mapping function between luminance and *perceptually uniform* (PU) coding that extends the usable range of luminance levels from regular 1-100 cd/m<sup>2</sup> up to 10<sup>6</sup> cd/m<sup>2</sup>. Since HDR displays are concerned in our study, the pixel luminance values should be converted into PU coding before applying any objective quality metric. Experimentally derived lookup table for mapping between luminance and PU luma values has been provided in [7]. The PU luma values have been scaled so that they provide close match with sRGB non-linearity in the range between 0.1-80 cd/m<sup>2</sup>.

The authors in [6] did not try to fit the PU mapping to an analytical function. However, such a function would be useful for conceptualizing the relationship between physical and perceptual luminance. This is why we are proposing an analytical function with three parameters, where the values of the parameters have been derived by fitting the function to the values in the lookup table with minimum least squares method. The resulting transfer function from luminance  $L$  to PU coding is given in Eq. (3), and the relationship curve is depicted in Fig. 3. For the purpose of comparison, traditional gamma correction function with  $\gamma=1/2.2$  is also shown.

$$PU(L) = 66.25 \cdot \ln(0.56 \cdot L^{0.88} + 1) \quad (3)$$

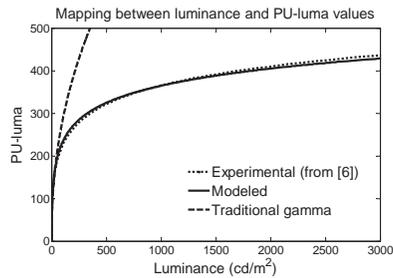


Figure 3. Mapping from luminance to perceptually uniform luma values.

Since PU luma extends the range of sRGB luma values beyond the original maximum, PU luma values need to be normalized in respect to the actual peak luminance for practical use. For this purpose, we define maximum target luminance  $L_{MAX}$ , that would respect to the maximum PU luma value  $PU_{MAX}$  (for example, when 8 bit coding of pixel values is used,  $PU_{MAX}=255$ ). The resulting scaled mapping is given in Eq. (4). The inverse function for converting PU luma values into physical luminance domain is given in Eq. (5). The experimentally derived values for the constants are  $a=0.56$  and  $b=0.88$ , respectively.

$$PU(L) = PU_{MAX} \cdot \frac{\ln(a \cdot L^b + 1)}{\ln(a \cdot L_{MAX}^b + 1)} \tag{4}$$

$$L(PU) = \left( \frac{\exp(PU / PU_{MAX}) \cdot \ln(a \cdot L_{MAX}^b + 1) - 1}{a} \right)^{1/b} \tag{5}$$

**3. MODELING BACKLIGHT DIMMING SYSTEM**

To evaluate the performance of different backlight dimming algorithms in terms of perceived image quality and power consumption without implementing the whole system in a real hardware and performing time consuming subjective quality assessment and power measurements, a software simulation modeling the physical characteristics of a display with local backlight dimming capability would be necessary. A block diagram of the proposed simulation environment is illustrated in Figure 4. The proposed model has some similarities with the model for global backlight dimming by Bartolini et. al. [8], the main difference being that we focus on local backlight dimming. Optimization of backlight dimming algorithms is out of the scope of this paper, and this is why backlight dimming is considered as a black box, subject to evaluation. However, for verification purposes, the basic algorithms and approaches for backlight dimming will be discussed in Section 3.2.

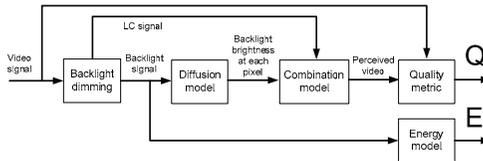


Figure 4. Model of a display with local backlight dimming capability.

**3.1 Functionality of the model**

In the proposed model, the backlight dimming block takes raw video in conventional digital format (either YUV or RGB) as input, and gives *RGB* signal for LC and backlight LED intensities *L* as output<sup>1</sup>. The relative energy consumption can be estimated directly from signal *L*, since the luminance intensity of a LED is supposed to be dependent on the energy consumption. The exact energy model depends on the physical characteristics of the LEDs and their control units, but in the sake of clarity, linear relationship between luminance and power consumption can be assumed with a reasonable accuracy. The diffusion model explained in Section 2, operates in the luminance domain and therefore takes the LED intensities as such as input. The output signal *A* contains the luminance domain brightness values of each pixel on the diffuser plate.

In the combination model, the luminance values *A* are converted to PU coded luma values *A'*<sub>PU</sub> using the transfer function given in Eq. (4). Basically, the perceived luma intensities of *R*, *G*, and *B* components of each pixel can be computed by multiplying *A'*<sub>PU</sub> values by the respective input *R*, *G* and *B* values that have been normalized to range 0-1. Some adjustments may be necessary, depending on the characteristics of the modeled display. Most notably, straightforward multiplication does not take light leakage into consideration. This is why we propose a combination rule given in Eq. (6), where *μ* is a light leakage factor that defines the proportion of light that leaks through an entirely black pixel (ie. *R=G=B=0*).

<sup>1</sup> Note that apostrophe refers to gamma corrected luma values, ie. *L'* is the luma domain value respective to physical luminance *L*.

$$RGB_{pu} = A'_{pu} \cdot \frac{RGB}{PU_{MAX}} + \mu \cdot \left( A'_{pu} \cdot \left( 1 - \frac{RGB}{PU_{MAX}} \right) \right) \quad (6)$$

In practice, the observed light leakage depends on the viewing angle. This is why different values of  $\mu$  could be used to model the distortion related to viewing position. Even if the viewer is positioned accurately in the front of the screen, the viewing angle would be slightly different to different parts of the screen. Therefore, the actual  $\mu$  would also be slightly higher in the edges than in the middle of the screen. However, this aspect is omitted in our study, and a constant  $\mu$  for the whole screen is assumed.

In the quality assessment phase, the combined signal representing the video perceived on the display is compared against the received video signal used as input to the backlight dimming algorithm. Naturally, if the original signal is in YUV format, the combined signal needs to be converted back to the same format, as well. In theory, any well established objective quality metric, such as PSNR or SSIM, could be used as a quality metric. However, the existing metrics have not been verified with test signals containing artifacts generated by backlight dimming, and this is why some subjective analysis would be needed to find out the most reliable metrics in this context.

### 3.2 Backlight dimming algorithms

In short, the goal of backlight dimming is to preserve the original intended brightness of each pixel with as low average luminance level of the backlight elements as possible. For this purpose, the first task is to compute a target luminance map, where each pixel is represented by the maximum pixel value of the R, G, and B components, converted into physical luminance by Eq. (5). It is usually not feasible to physically measure the luminance of each pixel during the backlight dimming procedure, and this is why the algorithm must rely on the simulated actual luminance map created by applying the backlight LED signal to the diffusion model. The luminance map has the same resolution as the display, and each element in the map describes the physical luminance of the respective pixel.

We have implemented a baseline algorithm for clipper-free backlight dimming, based on the basic idea described in [2]. The algorithm works iteratively. In the beginning, all the LEDs and the actual luminance map are initialized with zeros. For each iteration round, the most unsatisfied pixel (the pixel with the lowest luminance in respect to the target luminance) is identified. Then, the intensity of the closest LED to the pixel is increased to reduce the gap, and the actual luminance map is updated accordingly, using the diffusion model. If the maximum intensity of the LED is reached and the pixel is still unsatisfied, the algorithm searches the next closest LED and increases its intensity. This is repeated until the pixel becomes satisfied. Once all the pixels are satisfied, the algorithm terminates.

Optimally, the difference between the target luminance map and the actual luminance map should be as small as possible. In practice, there are almost always a lot of oversatisfied pixels, since each LED contributes a large amount of pixels in the vicinity and the pixels in a region may have very different target intensities. However, in most cases there will be clearly distinguishable darker and brighter areas in the luminance map. Since the original RGB signal assumes homogeneous luminance distribution on the diffuser, brightness preservation is needed to regain the original target intensities at each pixel. This is done by first converting the luminance map into normalized PU luma domain with range 0-1 using Eq. (4) and dividing by  $PU_{MAX}$ , and then computing the new pixel values  $T$  by dividing the intended pixel value  $I$  by the respective normalized backlight PU luma value  $A$  (see Eq. (1)).

In practice, the algorithm described above is computationally too complex and slow for real-time applications, such as frame-wise local backlight dimming for video sequences. This is why practical algorithms for local backlight dimming used for TV displays available in the consumer markets are based on suboptimal approaches, such as computing the histogram of pixel brightness levels in different regions and then dimming the respective backlight segments heuristically [9,10]. This kind of algorithms may give satisfactory results for images consisting of clearly distinctive bright and dark areas, but with more demanding images the result is typically suboptimal in respect to the quality, power saving, or both.

## 4. PRACTICAL EXPERIMENTS

In order to verify our conceptual approach of modeling a local backlight dimming system, we have run a range of experiments using a real display allowing local backlight dimming, manufactured by SIM2 [11]. The display has a 47 inch panel with full HD resolution (1920x1080 pixels). The backlight is composed of 2202 high power LEDs, arranged in a hexagonal grid. The LEDs can be controlled independently using the test program we have implemented for our experiments in C++.

In the first set of experiments, we studied the relationship between the actual light diffusion and the modeled light diffusion. The point spread function of an individual LED may be obtained by measuring the luminance at different points on the display when only one individual LED is turned on. We have used the point spread function provided by the display manufacturer instead of performing our own measurements. The point spread function can be converted to pixel values by transforming the luminance at each pixel into PU domain by Eq. (4). The real and the simulated light diffusion can be compared by displaying one illuminated LED in transparent LCD mode, and then displaying the normalized point spread function on LCD with all the backlight LEDs set to level that produces as uniform light distribution as possible, with luminance respective to the peak luminance of an individual LED.

In the following phase, we have validated the additive model for multiple point spread functions by comparing the actual co-impact of two closely located LEDs against simulated light diffusion generated by summing two point spread functions in respective positions in the luminance domain and then converting the resulting pixel values into PU domain. The visual analysis showed a reasonably accurate resemblance between simulated point spread functions.

In the third phase, we have used the baseline backlight dimming algorithm described in Section 3.2 to validate the full simulation model presented in Fig. 4. The resulting actual light diffusion pattern, simulated light diffusion pattern, image displayed on globally dimmed backlight and the same image displayed after local backlight dimming and brightness preservation are shown in Fig. 5. The images have been taken of the display with a digital camera.

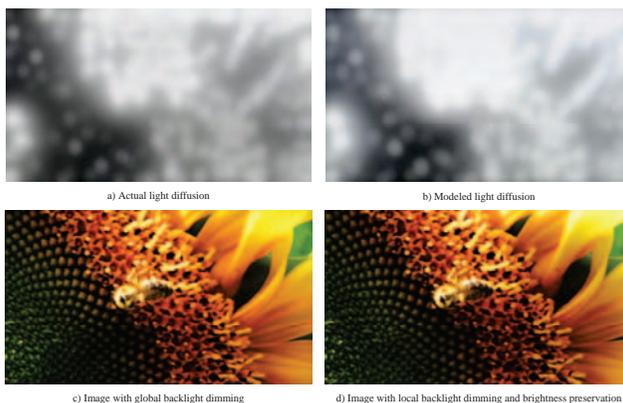


Figure 5. Comparison of actual (a) and modeled (b) light diffusion, and original picture on global backlight (c) and with local backlight dimming and brightness preservation (d).

We have repeated the backlight dimming procedure with different test images, representing different average brightness levels and visual characteristics. 'Sunflower' has a relatively complex texture and large contrast between the bright area in the upper right part and the darker left part of the image. 'Pedestrian' is a relatively dark image with some brighter areas. 'Blue Sky' has a large smooth bright blue surface partially shadowed by leaves of trees, producing a strong local contrast in those areas. 'Riverbed' is an image with small bright and dark areas forming a rather complex texture. The images with the average pixel value indicating the overall brightness are shown in Figure 6.



Figure 6. Test images and the respective average pixel values (in range 0-255) describing the average brightness.

According to the visual examination of the resulting images, the visual appearance with local backlight dimming and brightness preservation is close. The average LED intensities with the local backlight dimming in respect to the global backlight intensity (scaled to the same maximum luminance) are 40.3% (Sunflower), 25.7% (Pedestrian), 86.4% (Blue Sky) and 45.9% (Riverbed). Since a linear relationship between LED intensity and power consumption is assumed, the values describe the relative power consumption obtained by using local backlight dimming in respect to global dimming. As the results show, the relative power saving depends heavily on the content of the picture in question. The potential power saving cannot be predicted straightforwardly from the average pixel value, since the distribution of bright pixels in the image plays an even more significant role. The best power saving can be obtained for relatively dark images, where the bright pixels are strongly clustered in certain areas.

The practical experiments showed that even if the pre-computed map for the spreading function is used in the diffusion model, the process of updating the luminance map iteratively in our baseline backlight dimming algorithm is rather slow. Finding the optimal LED intensities takes up to several minutes for one image with the display used in our experiments. Of course, the display has an exceptionally large number of individually adjustable backlight LEDs (2202). This may be used for high quality image display, but in any case, it is probable that even with smaller amount of backlight segments the speed of the algorithm is not acceptable for real-time video applications. Optimization of backlight dimming is out of the scope of this paper, but the results obtained by the algorithm are conjectured to be close to optimal, and therefore they can be used as a comparison point when fast suboptimal backlight dimming algorithms are evaluated and developed.

## 5. CONCLUSIONS

In this paper, we have proposed a framework for assessing the performance of local backlight dimming algorithms in a simulated environment. The design target is to model the diffusion of backlight on the diffuser plate and the transmittance of pixel elements on the LC layer as accurately as possible, in order to create a numerical presentation of the displayed image that closely resembles the physical image on the screen as perceived by the viewer. Having established this, the traditional objective quality metrics can be used to evaluate the quality distortion between the original digital image and the displayed image created by the model, as well as comparison of different display solutions, e.g. different local backlight dimming algorithms. The major design challenges lie in accurate modeling of backlight diffusion and light leakage through LCs, since these properties are highly dependent on the actual display hardware. However, we have shown via practical experiments that the proposed approach can be successfully used to approximate the physical properties of a real display implementing local backlight dimming.

In order to validate our approach, we have implemented a model for backlight diffusion and an iterative baseline local backlight dimming algorithm to analyze the model offline. The visual analysis of the obtained images and light diffusion patterns shows that the model can be used for realistic simulation of light diffusion, which in turn can be used to evaluate the image quality and power consumption. The baseline algorithm used in this study is slow, but it produces results that are close to optimal in terms of image quality and power consumption, and therefore it can be used as a benchmark when suboptimal real-time algorithms are evaluated offline. In the future, our intention is to use the proposed framework in the research of backlight dimming algorithms suitable for practical applications.

## REFERENCES

- [1] Jang, T., "Ultra-Slim LCD TV Enabled by an Edge-Lit LED Backlight Systems," *Information Display*, 24(11), 14-17 (2008).
- [2] Albrecht, M., Karrenbauer, A., and Xu, C., "A Clipper-Free Algorithm for Efficient HW-Implementation of Local Dimming LED-Backlight," *Proc. IDRC'08, Orlando, FL, USA*, 286-289 (2008).
- [3] Oh, W-S., Cho, D., Cho, K-M., Moon G-W., Yang, B., and Jang, T. "A Novel Two-Dimensional Adaptive Dimming Technique of X-Y Channel Drivers for LED Backlight System in LCD TVs," *IEEE Journal of Display Tech.*, 5(1), 20-26 (2009).
- [4] An, J.Y., Kim, S-E., Song, W-J., Lee, T-W., and Kim, C-G., "Adaptive Local Dimming Backlight for Liquid-Crystal Displays," *Proc. IDW'08, Niigata, Japan*, 285-288 (2008).
- [5] Brooks, A., Zhao, X., and Pappas, T., "Structural Similarity Quality Metrics in a Coding Context: Exploring the Space of Realistic Distortions," *IEEE Trans. on Image Processing* 17(8), 1261-1273 (2008).
- [6] Aydin, T., Mantiuk, R., Seidel, H-P., "Extending Quality Metrics to Full Luminance Range Images," *Proc. SPIE 6806, San Jose, CA, USA* (2008).
- [7] [http://www.mpi-inf.mpg.de/resources/hdr/fullhdr\\_extension/](http://www.mpi-inf.mpg.de/resources/hdr/fullhdr_extension/)
- [8] Bartolini, A., Ruggiero, M., and Benini, L., "Visual Quality Analysis for Dynamic Backlight Scaling in LCD Systems," *Proc. DATE'09, Nice, France*, 1428-1433 (2009).
- [9] Chen, H., Sung, J., Ha, T., and Park, Y., "Locally Pixel-Compensated Backlight Dimming on LED-Backlit LCD TV," *Journal of the SID*, 15(12), 981-988 (2007).
- [10] Kim, S-E., An, J-Y., Hong, J-J., Lee, T. W., Kim, C. G., and Song, W-J., "How to Reduce Light Leakage and Clipping in Local-Dimming Liquid-Crystal Displays," *Journal of the SID*, 17(12), 1051-1057 (2009).
- [11] <http://www.sim2.com/HDR/>

## **A.2 Speedup of Optimization-based Approach to Local Backlight Dimming of HDR Displays**

**N. Burini**, E. Nadernejad, J. Korhonen, S. Forchhammer, and X. Wu,  
“Speedup of Optimization-based Approach to Local Backlight Dimming  
of HDR Displays”, in *Proceedings of SPIE*, vol. 8436, 2012, 84360B

Reference: [3].

## Speedup of optimization-based approach to local backlight dimming of HDR displays

Nino Burini<sup>a1</sup>, Ehsan Nadernejad<sup>a</sup>, Jari Korhonen<sup>a</sup>, Søren Forchhammer<sup>a</sup>, Xiaolin Wu<sup>b2</sup>

<sup>a</sup>Dept. of Photonics Eng., Tech. Univ. of Denmark, Ørstedts Plads, 2800 Kgs. Lyngby, Denmark;

<sup>b</sup>Dept. of Electrical & Computer Eng., McMaster Univ., Hamilton, Ontario, Canada, L8G 4K1

### ABSTRACT

Local backlight dimming in Liquid Crystal Displays (LCD) is a technique for reducing power consumption and simultaneously increasing contrast ratio to provide a High Dynamic Range (HDR) image reproduction. Several backlight dimming algorithms exist with focus on reducing power consumption, while other algorithms aim at enhancing contrast, with power savings as a side effect. In our earlier work, we have modeled backlight dimming as a linear programming problem, where the target is to minimize the cost function measuring the distance between ideal and actual output. In this paper, we propose a version of the abovementioned algorithm, speeding up execution by decreasing the number of input variables. This is done by using a subset of the input pixels, selected among the ones experiencing *leakage* or *clipping* distortions. The optimization problem is then solved on this subset. Sample reduction can also be beneficial in conjunction with other approaches, such as an algorithm based on gradient descent, also presented here. All the proposals have been compared against other known approaches on simulated edge- and direct-lit displays, and the results show that the optimal distortion level can be reached using a subset of pixels, with significantly reduced computational load compared to the optimal algorithm with the full image.

**Keywords:** Local backlight dimming, liquid crystal display, light emitting diode backlight, linear programming, optimization, high dynamic range display, gradient descent

### 1. INTRODUCTION

In this paper, we present techniques to reduce the complexity of optimization based algorithms for local backlight dimming in Liquid Crystal Displays (LCD). LCD is nowadays the most widespread display type, used for several kinds of devices, from mobile phones to 3D TV. In this study, we focus on Full High Definition (HD) LCD television using Light Emitting Diode (LED) backlight, displaying images and video. Since Liquid Crystals (LC) are basically voltage controlled light filters, an LCD requires a light source in addition to LCs. Certain types of LCDs, such as those used in digital watches, may rely on ambient light, but TV and computer displays usually contain a built-in backlight, located behind the LC layer.

Traditionally, the backlight designs aim to provide even distribution of light over the whole display area. This can be achieved by using a carefully designed diffuser plate between the backlight and LC. Backlight can be located either directly behind the diffuser, or at the edges of the screen. Conventionally, Cold Cathode Fluorescent Lamps (CCFLs) have been most commonly used as backlight, due to their inexpensive cost and reasonable energy efficiency. However, the backlights based on LEDs are now becoming a commonplace, thanks to the rapid advances in LED technology in terms of cost and power consumption. Due to the tightening regulations and increasing environmental awareness, energy efficiency is a crucial issue in the TV industry, and since the backlight is typically the most power consuming component of an LCD, power efficiency is a highly essential criterion for selecting the backlight technology.

<sup>1</sup> nibu@fotonik.dtu.dk; phone +45 4525 7374

<sup>2</sup> xwu@mail.ece.mcmaster.ca; phone +1 905 525-9140

One advantage of LEDs is that the light intensity can be changed in a flexible manner. This allows a technique called local backlight dimming: backlight can be consist of several segments, and brightness of each segment can be adjusted to match the content in the respective segment of the displayed image. If the image contains both dark and bright regions, significant power savings can be reached by using lower backlight intensity in the dark areas. Another benefit of local backlight dimming is the reduced light leakage. Due to imperfections of practical LCs, they are not capable of blocking all the light through black pixels. This makes black pixels look slightly grayish, which reduces the perceived contrast of the image. However, light leakage can be efficiently restricted by using low backlight level in the dark regions.

Different display architectures with backlight dimming capability exist. The most trivial form of backlight dimming with only one backlight segment covering the whole display is referred as *global backlight dimming*. In edge-lit displays, the screen may be divided in vertically or horizontally directed backlight segments, allowing so called 1D dimming. The backlights can also be allocated at both sides so that the 1D backlight segments are split in two parts; this would be called 1.5D dimming. Backlight segments located directly behind diffuser (direct-lit) allows most flexibility, and such an architecture is referred as 2D dimming. LCDs using different local backlight configurations, with the number of segments ranging from a few up to several thousands, have been built. Figure 1 shows examples of 2D and 1.5D backlight.

Apparently, local backlight dimming offers substantial potential benefits in the form of power savings and improved contrast, but there are also challenges. In practical backlight architectures, the light from different backlight segments is mixed on the diffuser, and the backlight luminance at each pixel position is therefore contributed by several backlight segments. This is why finding an optimal combination of backlight intensities is a very challenging optimization problem, basically including all the pixels and backlight segments as variables. Bright and dark pixels located close to each other are especially challenging. If the backlight level is too low for a bright pixel, the target intensity cannot be reached. In this case, the pixel is said to be *clipped*. On the other hand, when the backlight level is high, the dark pixels may suffer from *light leakage*. Leakage is often especially disturbing around a bright object, where a *halo* effect is observed. In most cases, a fully optimal solution cannot be found, but a trade-off between clipping and leakage must be chosen.

Because optimized local backlight dimming is a very complex computational problem, most practical implementations and solutions known from the related literature are based on suboptimal algorithms. Most of these proposals aim at optimizing the image quality, whereas power saving is seen more as a positive additional benefit. An optimization-based algorithm based on linear programming has been presented in our earlier work [1], later extended with a cost function where different weight can be applied to image quality and power consumption [2]. Unfortunately, the computational load of the proposed algorithm is too high for real-time applications, such as TV display. In this paper, we present approaches for reducing the complexity of the proposed approach, while maintaining the optimality as well as possible.

The rest of this paper is organized as follows. In Section 2, the basic concepts and approaches for modeling a local backlight dimming system are explained. In Section 3, we explain the optimization-based backlight dimming algorithm and the proposed extensions for speeding up the algorithm. In addition, we explain selected algorithms from the literature, used as a comparison point in the experimental part of our work. Section 4 describes the practical experiments we have performed to validate the proposed algorithms and summarizes the results of the experiments. Finally, the concluding remarks are given in Section 5.

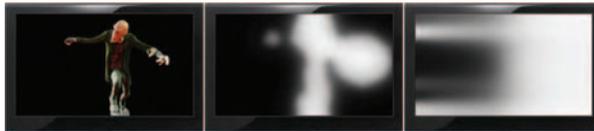


Figure 1: A frame displayed on a LCD and examples of direct-lit "2D" backlight and edge-lit "1.5D" backlight.

2. MODELING

In order to implement any practical backlight dimming algorithm, it is first necessary to model the essential characteristics of a display with local backlight dimming capabilities. The most essential concepts are *transmittance*, that defines the proportion of light an LC pixel is adjusted to let through, and *backlight level*, that defines the brightness level of a backlight element. These are the input signals to LCD. When a color display is concerned, each pixel actually contain three subpixels: red (R), green (G) and blue (B). For the sake of simplicity, we only consider a single color channel in this work, but the presented equations and formulations can be applied to all the three color components. The observed backlight intensity at certain pixel position depends on the physical structure of the diffuser plate. The distribution of light coming from a single element is modeled using a *Point Spread Function* (PSF), and the contributions from different light sources need to be summed up to model the total observed backlight at each pixel. Because of light leakage, there may also be a mismatch between intended transmittance and observed transmittance. All these issues need to be considered in the backlight model.

2.1 Transmittance and leakage

The observed luminance  $L$  of pixel  $(i, j)$  in a locally dimmed backlight LCD screen can ideally be expressed as the product of the backlight intensity  $B$  and the LC transmittance  $T$ :

$$L(i, j) = B(i, j)T(i, j). \tag{1}$$

The values of  $L$ ,  $B$  and  $T$  are all normalized to the interval  $[0, 1]$ , so  $B(i, j)=0$  means that there is no light behind pixel  $(i, j)$ , while  $B(i, j)=1$  means that the intensity of the light is at its maximum. Similarly for the transmittance,  $T(i, j)=1$  means that all the backlight is passed through LC, while  $T(i, j)=0$  means that the pixel is fully blocked. Ideally,  $T$  is the same as the driving signal  $\gamma$  controlling LCs. However, in practice, LC cannot block all the light, due to leakage. Leakage can be modeled linearly by using a parameter  $\varepsilon$ , also called leakage factor, defined as the amount of light leaking through pixel  $(i, j)$  when  $T(i, j)=0$  and  $B(i, j)=1$ . The model of output luminance  $L$  in the presence of leakage is given as

$$L(i, j) = B(i, j)T(i, j) + \varepsilon(i, j)B(i, j)(1 - T(i, j)) \tag{2}$$

or alternatively:

$$L(i, j) = B(i, j)T_o(i, j), \tag{3}$$

where  $T_o$  is the observed transmittance, as opposed to ideal transmittance  $T$ .  $T_o$  can be expressed as

$$T_o(i, j) = (1 - \varepsilon(i, j))T(i, j) + \varepsilon(i, j). \tag{4}$$

The leakage factor  $\varepsilon$  can be different for each pixel, depending on the pixel position and the viewing angle, for example. For simplicity, a constant  $\varepsilon$  across the whole screen is assumed in this paper. Figure 1. illustrates the model graphically.

2.2 Backlight diffusion

Light diffusion in the backlight can be expressed as a function of the intensities of the backlight segments, and their Point Spread Functions (PSFs). The backlight resulting from a certain array of LED values at the pixel  $(i, j)$  is simply the linear combinations of all the PSFs multiplied by the corresponding LED intensity, given as

$$B(i, j) = \sum_{k=1}^N r_k h_k(i, j), \tag{5}$$

where  $B(i, j)$  is the backlight at pixel  $(i, j)$ ,  $N$  is the number of backlight segments,  $r_k$  is the intensity of the  $k$ -th segment and  $h_k(i, j)$  is the value of the PSF of the  $k$ -th segment to pixel  $(i, j)$ . It is possible to express Eq. (5) in matrix form:

$$\mathbf{b} = \mathbf{H}\mathbf{r}, \tag{6}$$

where  $B(i,j)$  values are represented by the column vector  $\mathbf{b}$  with as many elements as there are pixels, PSFs are represented by the *influence matrix*  $\mathbf{H}$  with  $N$  columns and a row per pixel, and  $\mathbf{r}$  is a column vector with  $N$  backlight values.

There are cases where it is unpractical or not possible to know all the PSFs. For example, in the case of a direct-lit backlight with a high number of segments, a large amount of memory would be required to store all the PSF data. One solution is to use the same PSF for all the segments; this PSF can be obtained by averaging over some PSFs from different segments, for example. The downside of this approach is that the segments close to the edges of the screen will trim the PSF and modeling of light diffusion around the edges becomes inaccurate.

A common assumption is that backlights are designed so that if all the LEDs have the full intensity, the resulting full backlight would be smooth and uniform. With this assumption, it is possible to simulate the same effect by scaling the PSFs by the full backlight:

$$h'_k(i, j) = h_k(i, j)B_r(i, j), \quad (7)$$

where  $B_r(i,j)$  is the full backlight at pixel  $(i,j)$ , assuming that all the backlights have full intensity:

$$B_r(i, j) = \sum_{k=1}^N h_k(i, j). \quad (8)$$

### 2.3 Backlight-pixel interaction and brightness compensation

When the backlight is dimmed, the consequent reduction of luminance can be compensated by increasing the transmittance of the LC pixel. This step is usually referred as *pixel compensation* [3]. In an ideal case, the compensated transmittance  $T_c$  can be solved from Eq. (1) by replacing  $L$  with the target image  $L_t$ :

$$T_c(i, j) = \frac{L_t(i, j)}{B(i, j)}. \quad (9)$$

However, in practice  $T_c$  has a limited range of valid values. In particular, it is up-bounded at 1, because otherwise it would mean that LCs could amplify the backlight luminance. On the other hand, observed transmittance is low-bounded at  $\epsilon$ , because of the leakage. To set the upper bound and compensate the impact of leakage, Eq. (9) should be rewritten

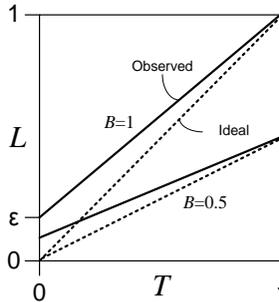


Figure 2. Impact of leakage factor  $\epsilon$  on output luminance  $L$  with ideal transmittance  $T$ , when the backlight intensity is  $B$ .

as:

$$T_c(i, j) = \begin{cases} 1 & \text{if } \frac{L_v(i, j)}{B(i, j)} > 1 \quad (\text{Cond. I}) \\ 0 & \text{if } \frac{L_v(i, j)}{B(i, j)} < \varepsilon \quad (\text{Cond. II}) \\ \frac{L_v(i, j) - \varepsilon}{1 - \varepsilon} & \text{else} \quad (\text{Cond. III}) \end{cases} \quad (10)$$

Pixels for which Condition I is true will experience *clipping*, which means that the LCs cannot compensate properly the backlight reduction. As a result, the pixels looks dim. On the contrary, pixels for which Condition II is true will show *leakage* and look brighter than they should. These pixels can be called *clipped* and *leaking*, respectively. After solving  $T_c$ , the observed physical transmittance can be computed from Eq. (4), by replacing  $T$  with  $T_c$ .

#### 2.4 Perception of brightness

Since the Human Visual System (HVS) perceives luminance non-linearly, the perceived severity of a luminance error on a dark pixel is different from the error of a similar magnitude on a bright pixel. In fact, the HVS is more sensitive to luminance changes at low luminance levels than at high levels. Because of this, the impact of leakage on subjective image quality is larger than indicated by the physical luminance error. The sensibility to luminance changes decreases, as the luminance increases. The perceived response is often approximated with a power function of the form:

$$L_v = L^\gamma, \quad (11)$$

where  $L_v$  is the perceived luminance and  $\gamma$  is the Gamma. Gamma value of 2.2 is typical, but other values have been used as well [3]. The inverse response can simply be calculated:

$$L = L_v^{1/\gamma}. \quad (12)$$

$L_v$  is said to be perceptually uniform, ie. a fixed step always indicates similar perceived difference, whereas the same steps correspond to different intervals of variable length for  $L$ . Conversely, equal steps of  $L$  correspond to variable step sizes of  $L_v$ , determined by the response (Gamma) function. We say that  $L_v$  represents luminance in perceptual domain and  $L$  in physical domain. It should be noted that the model described in this Section operates in the physical domain, ie. physical luminance is assumed in Eqs. (1)-(10).

The use of Gamma to approximate the HVS response to luminance is based on the assumption that the peak luminance of the display is relatively low, about 100 cd/m<sup>2</sup>. This is not a valid assumption for advanced High Dynamic Range (HDR) LCD displays, that can have peak luminance values up to 4000 cd/m<sup>2</sup>, like one of the screens modeled for the experiments reported in this work [4]. For this reason, other response functions have been proposed [5]. However, in this work the traditional Gamma function is adopted for clarity.

### 3. ALGORITHMS

This Section presents the optimization-based backlight dimming algorithm in [1], along with a possible strategy to reduce the complexity of the problem and an alternative approach based on Gradient Descent optimization. The last sub-Section describes other backlight dimming algorithms known from the literature, used for the purpose of comparison.

### 3.1 Optimization-based algorithm

Let  $\mathbf{y}$  be a vector representing the ideal target image of size  $m \times n$ , and  $\mathbf{x}$  a vector representing the actual image rendered on the display, respectively, it is possible to formulate backlight dimming as an optimization problem [1], as shown below:

$$\begin{aligned} & \text{minimize} && \|\mathbf{y} - \mathbf{x}\| \\ & \text{subject to} && \mathbf{x} = \mathbf{t} \circ \mathbf{b} \\ & && \mathbf{b} = \mathbf{H}\mathbf{r} \\ & && \varepsilon \leq t_i \leq 1, \quad i = 1, \dots, mn \\ & && 0 \leq r_i \leq 1, \quad i = 1, \dots, N \end{aligned} \quad (13)$$

where  $\mathbf{x}$  is the product of the LC transmittances  $\mathbf{t}$  (vector presentation of observed transmittances) and the backlight  $\mathbf{b}$  (see Eq. 3), which depends on the LED values  $\mathbf{r}$  and the PSF information contained in the influence matrix  $\mathbf{H}$  (see Eq. 6). Because of leakage, the observed transmittance of the LCs is low-bounded by the leakage factor  $\varepsilon$ , see Eq. (10). The LED values range between zero (minimum, turned off) and one (maximum, full power). In Eq. (13), the aim is to minimize the cost function based on the distance between  $\mathbf{y}$  and  $\mathbf{x}$ . The cost can be calculated in 1-norm, 2-norm or another norm. As shown in [1], it is possible to reformulate the problem as a convex problem. Using 1-norm the problem is linear, and with 2-norm it is quadratic.

It is possible to generalize Eq. (13) by including a term in the cost function favoring solutions with lower power consumption and introducing error weighting for each pixel of the  $\mathbf{y}$  and  $\mathbf{x}$  difference [2]. These extensions are included in the formulation that follows:

$$\begin{aligned} & \text{minimize} && \left\| (\mathbf{y} - \mathbf{x}) \circ \mathbf{w} \right\| + q \cdot p \\ & \text{subject to} && \mathbf{x} = \mathbf{t} \circ \mathbf{b} \\ & && \mathbf{b} = \mathbf{H}\mathbf{r} \\ & && \varepsilon \leq t_i \leq 1, \quad i = 1, \dots, mn \\ & && 0 \leq r_i \leq 1, \quad i = 1, \dots, N \end{aligned} \quad (14)$$

where  $\mathbf{w}$  is the error weighting vector,  $p$  is a measure of power consumption (in this work we use the average LED value) and  $q$  a weight of its impact in the cost function. It is clear that if  $\mathbf{w}$  is a vector of ones and  $q$  is equal to zero, Eq. (14) is equivalent to Eq. (13).

This optimization-based algorithm can find the optimal solution minimizing the error for one image on a specific backlight configuration. In its original form, the optimization is done in the physical luminance domain, as opposed to the perceptual domain, which means that the algorithm minimizes the luminance error for each pixel. However, due to the non-linear nature of luminance perception, it is not possible to do linear optimization in the perceptual domain; it is however possible to use the weighting vector  $\mathbf{w}$  in Eq. (14) to give a different weight to errors, depending on the target luminance of each pixel. The weights can be derived from the slope of a curve modeling the human response to luminance at the target luminance level, for example.

### 3.2 Proposals for reduction of complexity

The optimization-based approach can find the optimal backlight for a given image. However, in this case the complexity increases very rapidly with the number of variables; considering that Full HD screens have more than two million pixels and each pixel corresponds to a variable, this is obvious. It is thus important to consider strategies that allow solving the problem with a reduced set of variables.

One possible solution is to downscale the input image, solve the problem at the reduced resolution, and finally apply the resulting solution to the full resolution image [1]. The efficacy of this approach depends on the downscaling technique

and factor. The problem with downscaling is that it either causes aliasing or uses a low pass filter, blending very bright or very dark pixels and therefore resulting a larger number of clipped or leaking pixels.

Another strategy is to find the optimum over a subset of the pixels from the original image. We propose to solve the problem on the distorted pixels of the image, i.e. the pixels experiencing clipping or leakage, see Eq. (10). It can be said that in backlight dimming the very bright pixels and the very dark pixels are the most challenging, for two reasons: firstly, the two are in conflict with each other, particularly if they are located close to each other, because a proper backlight level for the pixels in one group might mean leakage or clipping for the pixels in the other group; secondly, all the other pixels with intermediate target intensity can be rendered properly with a wider range of backlight intensities. This means that the only pixels contributing to the cost function are those that cannot be properly compensated, that is, the distorted leaking and clipped pixels. This is why it is a reasonable conjecture that the optimal solution for an image could be reached by solving the optimization problem on those distorted pixels only.

If an optimal backlight solution is found on a subset of pixels and this solution generates no new distorted pixels, then it is the overall optimal solution. In order to utilize this observation, we propose the following approach. Given an image and backlight intensities provided by any algorithm, all the leaking and clipped pixels of the image are listed. After compensation, the leaking pixels have a luminance level higher than the target luminance, while the clipped pixels have a luminance level lower than the target luminance. Once this subset of distorted pixels is selected, the optimization problem can be solved on it, and the resulting solution can be applied on the full image. All the pixels that are distorted after this step, but were not distorted in the previous step, are added to the list. Then, the algorithm is re-iterated, until there are no new distorted pixels. The cost function to minimize can then be formulated as follows: let  $D_c$  denote the set of pixels distorted by clipping and  $D_l$  the pixels distorted by leaking, respectively. Using 1-norm, the cost function in Eq. (15) may then be expressed as:

$$f = \sum_{(i,j) \in D_c} (y(i,j) - B(i,j)) + \sum_{(i,j) \in D_l} (\epsilon B(i,j) - y(i,j)), \quad (15)$$

where  $y(i,j)$  is the target image value at position  $(i,j)$ .

This approach reduces the number of variables used in each iteration. However, the additional outer loop of iteration rounds increases the execution time. Nevertheless, experimental results show that it is possible to find nearly optimal backlight levels without using all the pixels from the original image.

The presented approach is just one alternative. The subset of pixels could be selected in many other ways, depending on the chosen criteria. It is possible to use segmentation techniques, for example, or perform histogram analysis to identify the most crucial pixels. These techniques will be examined in our future work.

### 3.3 An approach based on gradient descent

Given a specific cost function, optimizing the optimal backlight for an image  $y$  can also be approached through gradient descent algorithm, that is a search-based iterative strategy. The solution space has a number of dimensions, equal to the number of backlight segments. For a given solution, it is possible to calculate the derivative of the cost function over all the dimensions and thus find the gradient. The next solution is then obtained by adjusting the solution in the direction of the gradient. This process can be iterated for any given number of steps, or until a terminating condition is met (for example, when the optimum is reached). The solution after each iteration is expressed as:

$$\mathbf{r}_{i+1} = \mathbf{r}_i - s \nabla f(\mathbf{r}_i), \quad (16)$$

where  $\mathbf{r}_i$  is the solution at step  $i$ ,  $f$  is the cost function for a given target image  $y$ , and  $s$  is the step size. The minus sign is used because in this case the aim is to minimize the cost. The length of step  $s$  can be varied for each iteration; large steps allow quick convergence, while shorter steps are more precise and return lower cost solutions. In our initial implementation of the search based approach, we opted for a variable step size: first, the solution is found iterating with a relatively large step size, then the step size is halved, and the search starts again from the new solution; the process is then iterated, until the step size becomes smaller than a predefined threshold.

Assuming that the cost function is based on 1-norm and  $B(i,j)$  is expressed by Eq. (5), the partial derivative of the cost function in Eq. (15) with respect to  $r_i$  gives:

$$\frac{\partial f}{\partial r_k} = - \sum_{(i,j) \in D_c} h_k(i,j) + \varepsilon \sum_{(i,j) \in D_L} h_k(i,j). \quad (17)$$

If there are pixel values,  $y(i,j)$ , at the clipping boundary  $y(i,j) = B(i,j)$  or the leakage boundary  $y(i,j) = \varepsilon B(i,j)$ , the partial derivatives given by Eq. (17) are not strictly well defined, as the contributions to the cost function are non-differentiable, although both the right and left derivatives are well defined.

This iterative strategy is particularly flexible, as it allows to improve existing solutions easily. For example, assuming a video sequence, where consecutive images are very similar and have expectedly similar optimal backlights, the gradient descent based search can be used to calculate the solution for the current frame starting from the solution for the previous frame. It is also possible to use this approach on a reduced set of pixels, as described in the previous subsection; in this case, the gradient might be calculated over the distorted pixels only.

### 3.4 Algorithms for comparison

In order to compare our approaches against the other proposed solutions, we have selected four backlight dimming algorithms presented in the literature. The selected algorithms are described below.

Albrecht et. al. [6] introduced a clipper free algorithm, which under this constraint minimizes the power consumption. This formulation may be seen as a limiting case of Eq. (14) by setting the leakage to 0, let the power weight,  $q$ , go to 0. For practical implementation they suggest a clipper-free algorithm consisting of three steps. In the first step, the lower bounds are set for each backlight segment, depending on the image content and the PSF. The second step is optional and iterative: during each round, the most unsatisfied pixel, i.e. the pixel that requires the largest increase in luminance to be rendered properly, is found and the most influential LED for this pixel is increased to the intensity to satisfy the pixel in question. If the LED is already at its maximum, then the second most influential LED is used; if the second is at its maximum, then the third most influential LED is enhanced, and so on. The process is converged, when all the pixels are satisfied. The final third step scans the pixels of each segment in a specific order determined by the PSF and adjusts the LED values to make sure that every pixel receives enough backlight.

The algorithm introduced by Cho et. al. [7] uses the relationship between the average and the maximum luminance of the input image to calculate the backlight luminance by adopting an additional correction term. The backlight luminance is calculated by:

$$r_k = \text{avg}(\mathbf{y}_k) + \text{corr}, \quad (18)$$

$$\text{corr} = 0.5 \times \left( \max(\mathbf{y}_k) - \text{avg}(\mathbf{y}_k) + \frac{(\max(\mathbf{y}_k) - \text{avg}(\mathbf{y}_k))^2}{2^n} \right), \quad (19)$$

where  $r_k$  is the backlight luminance of element  $k$ ,  $\mathbf{y}_k$  contains the original pixels in backlight segment  $k$ , and  $n$  is the bit depth of  $\mathbf{y}$ . The values of the correction value *corr* can be stored in a look-up-table.

In [8], Nam introduces a low power local dimming algorithm described in the following steps:

*Step 1:* The mean value  $m_k$  of the average and maximum pixel values for a given segment  $k$ , corresponding to the respective backlight segment, is determined as follows:

$$m_k = \frac{\max(\mathbf{y}_k) + \text{avg}(\mathbf{y}_k)}{2}. \quad (20)$$

A roll-off point value  $m_{full}$ , calculated in a similar fashion from the full image, is assigned to all blocks.

*Step 2:* For the full image, the backlight luminance is calculated as follows:

$$r_{full} = r_i \left( \frac{m_{full}}{2^n} \right)^{\gamma}, \quad (21)$$

where  $r_i$  is the initial backlight luminance without active dimming, and  $\gamma$  is the gamma value of a given display which deals with 8-bit image data (ie,  $n=8$ ).

*Step 3:* If  $m_i$  is bigger than  $m_{full}$ , its backlight luminance is set as for the full image in Step 2. Elsewhere,  $\max(\mathbf{y}_k)$  is compared with  $\text{avg}(\mathbf{y}_{full})$ . When  $\max(\mathbf{y}_k)$  is less than  $\text{avg}(\mathbf{y}_{full})$ , the backlight luminance is computed from Eq. (22). If  $\max(\mathbf{y}_k)$  is larger than  $\text{avg}(\mathbf{y}_{full})$ , the dimming algorithm is expressed by Eq. (23).

$$r_k = r_i \left( \frac{\max(\mathbf{y}_k)}{2^n} \right)^\gamma, \tag{22}$$

$$r_k = r_i \left( \frac{\text{avg}(\mathbf{y}_{full})}{b} \right)^\gamma, \tag{23}$$

where

$$b = 2^n \cdot \left( 1 - \frac{\text{avg}(\mathbf{y}_{full})}{m_{full}} \cdot \left( \max(\mathbf{y}_k) - \text{avg}(\mathbf{y}_{full}) \right) \right) \tag{24}$$

Finally, the Max algorithm [9] sets the intensity of each backlight segment to the maximum value of the pixels contained in the respective segment.

### 4. EXPERIMENTAL RESULTS

This Section reports the results of some experiments that have been run to test the performance of the proposed solutions. Each subsection contains a description of the experiments, the relevant results and the following comments.

The experiments have been run for 32 different input images. Of these, 24 belong to the Kodak True Color Image Suite [10], the remaining include two images of high-contrast synthetic content, two compressed pictures acquired with a digital camera, two video frames and two natural images. All images have been rescaled to Full HD resolution with bicubic interpolation. For simplicity, the color images have also been converted to grayscale with the *rgb2gray* function from Matlab. Finally, normalized grayscale values have been linearized by elevating them by the power of  $\gamma = 2.2$ .

The displays with backlight dimming capabilities have been simulated. The first one is based on an edge-lit display with 1.5D backlight having 16 segments placed in 8 rows and 2 columns; the PSFs have been measured with a digital camera and properly post-processed. The second display is based on a 47" local backlight dimming Full HD screen manufactured by SIM2 [4]; this screen has 2202 backlight segments placed in a hexagonal grid and the PSF has been



Figure 3: The set of images used in the experiments.

provided by the manufacturer. The leakage factor  $\epsilon$  has been set to 0.001 in all simulations, relative to a peak luminance of 1.

To reduce computational requirements and processing time of optimization, the input images have been downsampled by a factor 10 for each dimension. The resulting solutions have then been applied to the original image at full resolution and an inverse  $\gamma$  has been applied to the normalized result to return to a perceptually uniform representation. This introduces a form of distortion when the results of the algorithms are evaluated at full resolution.

Distortion has been measured at two levels: on the low resolution linearized image, where the optimum has been found, and on the high resolution perceptually uniform image, which can be compared against the input grayscale image and represents what the screen displays.

In this work, optimization is performed in the physical domain, where it is possible to formulate the problem linearly. Ideally, however, optimization should be done in the perceptual domain, but this problem cannot be formulated linearly. This introduces a form of distortion which can be noticed when measurements are done in the perceptual domain, since the optimal solution in the physical domain does not usually coincide with the perceptual optimum.

One last note about distortion is that the results reported here allow transmittances and LED intensities to have any value between zero and one. However on real systems these values need to be quantized depending on the native bit-depth of the LC signals and of the LED intensities; the most common case is a precision of 8 bits, but HDR system for professional systems can have precisions as high as 12 bits.

#### 4.1 Sample reduction through detection of distorted pixels

This first experiment tested the performance of the complexity reduction strategy proposed in Section 3.2. This solution and the original method have been simulated on the two displays that have been modeled, the SIM2 screen and the other one with 1.5D backlight. The input images have been downsampled by a factor 10 and by factor 4 and the minimization has been performed using both 1-norm and 2-norm. The starting solution used by the proposed methods has been generated with the Max algorithm (see 3.4). Mean Absolute Error (MAE) and Mean Squared Error (MSE) have been measured at the low resolution physical level, while PSNR was calculated at the high resolution perceptually uniform level; power consumption was also considered, as well as the average number of iterations required for convergence of the proposed approach and the average percentage of samples from the original problem that was used in the last iteration. The results are presented in Table 1.

Table 1: Comparison of the proposed sub-sampling strategy ("Subset") in 1-norm (a) and 2-norm (b) optimization.

	Downsampled by 10				Downsampled by 4			
	1.5D		SIM2		1.5D		SIM2	
	Original	Subset	Original	Subset	Original	Subset	Original	Subset
Avg. LED value	82.36%	82.89%	73.06%	67.12%	87.91%	89.30%	81.21%	75.77%
Avg. MAE	$3.78 \cdot 10^{-5}$	$3.78 \cdot 10^{-5}$	$1.05 \cdot 10^{-5}$	$1.05 \cdot 10^{-5}$	$4.62 \cdot 10^{-5}$	$4.62 \cdot 10^{-5}$	$1.81 \cdot 10^{-5}$	$1.81 \cdot 10^{-5}$
Avg. iterations	-	2.59	-	2.67	-	2.69	-	2.91
Subsampling % in last iteration	-	9.70%	-	25.70%	-	9.55%	-	17.73%

a

	Downsampled by 10				Downsampled by 4			
	1.5D		SIM2		1.5D		SIM2	
	Original	Subset	Original	Subset	Original	Subset	Original	Subset
Avg. LED value	76.50%	84.66%	48.80%	61.03%	87.82%	92.62%	57.72%	67.51%
Avg. MSE	$4.04 \cdot 10^{-8}$	$4.04 \cdot 10^{-8}$	$8.70 \cdot 10^{-9}$	$8.70 \cdot 10^{-9}$	$5.73 \cdot 10^{-8}$	$5.73 \cdot 10^{-8}$	$1.48 \cdot 10^{-7}$	$1.48 \cdot 10^{-7}$
Avg. iterations	-	2.59	-	2.59	-	2.72	-	2.97
Subsampling % in last iteration	-	9.70%	-	25.86%	-	9.48%	-	17.95%

b

In all the considered cases, the proposed approach reaches virtually the same 1-norm or 2-norm quality if compared to the original algorithm. Convergence occurs in about 2 or 3 iterations in average, and the last iteration averagely used about 9.5% of the original pixels in the case of the 1.5D screen and between 18% and 26% in the case of the SIM2 screen.

The results show that it is possible to obtain quasi-optimal solutions with a significant complexity reduction. Nevertheless, it should be reminded that the results depend on the starting solution that has been used, in this case the one produced by the Max algorithm, and that although the proposed approach reduces the number of variables in the optimization problem it also introduces iteration, which increases overall complexity.

**4.2 Performance of the gradient descent based approach**

The gradient descent based approach proposed in Section 3.3 has been compared against Eq. (13). The starting solution has been calculated with the Max algorithm (see 3.4) and a maximum of 1000 iterations was allowed. Both 1-norm and 2-norm optimization have been tested on the 1.5D backlight screen model, with the input images scaled by a factor 10. The results for 1-norm and 2-norm optimization are shown in Table 2.

MAE and MSE have been measured at low resolution in the physical domain. In both cases it is possible to get very close to the optimal solution. Convergence took in average 95.53 iterations for 1-norm and 812.69 for 2-norm. Convergence is faster for 1-norm than it is for 2-norm, however it should be remarked that the most relevant improvements in MAE or MSE tend to happen in the first steps. To show this, the same experiment has been run again but with a limit of 10 iterations for 1-norm and of 50 iterations for 2-norm. In the first case, convergence occurred in average after 5.44 steps with an average MAE of  $3.83 \cdot 10^{-5}$ , which corresponds to 99.85% of the possible improvement from the starting solution to the optimum; in the 2-norm case, the average number of steps was 40.81 and the average MSE was  $5.45 \cdot 10^{-3}$ , corresponding to 99.96% of possible improvement. Figure 4 shows the average MAE decrease in the first 10 steps on gradient descent 1-norm minimization; about 95% of the possible improvement happens after the first

Table 2: Performance of the gradient descent based proposed solution for 1-norm and 2-norm.

	Optimal	Subset	Gradient Descent	Max	
MAE ( $\cdot 10^{-5}$ )	3.78	3.78	3.81	37.3	1-norm minimization
MSE ( $\cdot 10^{-9}$ )	4.04	4.04	4.08	4041	2-norm minimization

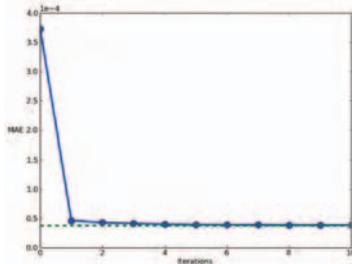


Figure 4: Average MAE reduction in the first step of 1-norm minimization with gradient descent; the dashed line represents the optimum.

step. This makes search based techniques like this one particularly interesting in the case of backlight dimming of video sequences, where the backlight solution for one frame can be used as starting solution for the following ones and time and hardware limitations might allow only a limited number of iterations.

It should be noted that the efficacy and the efficiency of such techniques are affected by several aspects like the choice of the starting solution and the determination of the variable step size. The proposals reported here might not be the optimal ones, however these first results show the potential of search strategies applied to backlight dimming.

#### 4.3 Introduction of power penalty in the cost function

The original optimization based algorithm, see Eq. (13), does not include power consumption in the cost function. However, it is considered in the extended version given by Eq. (14). The goal of this experiment is to evaluate the effect of the proposed sub-sampling strategy when power consumption is penalized. Following the notation used in Eq. (14),  $p$  was calculated as the average value of the intensity of the backlight segments, which is an approximation of the normalized power consumption of the backlight; the weight  $q$  was instead set to 0, 1, 10, and 50 for 2-norm optimization. With these parameters, the extended algorithm was run for the 1.5D backlight screen model with the input images downsampled by a factor 10. The distortion at the low resolution physical/linearized level was measured in MSE. The results are shown in Figure 5, where they have been compared against those from other algorithms. The sub-sampling based algorithm always achieves nearly-optimal quality levels; the same applies to the power consumption, except for  $q=0$  where it has increased.

The power weighting extension introduced in Eq. (13) allows to identify the optimum at several power levels and to compare the optimization based algorithms against other solutions that have been proposed, as shown in Figure 5.

#### 4.4 Impact of error weighting

In this experiment, we have introduced an error weight to the optimization problem (Section 3.2). The error weight is

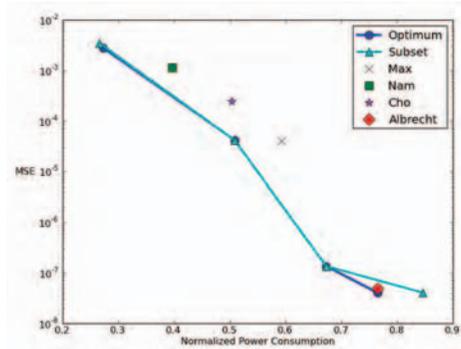


Figure 5: Impact of power penalization on 2-norm minimization on the optimization based algorithm and on the "Subset" sub-sampling approach.

different for each pixel  $(i, j)$  and is determined by [2],

$$w(i, j) = y(i, j)^{\gamma}, \tag{25}$$

where  $y(i, j)$  is the target luminance of the optimization for pixel  $(i, j)$  and  $\gamma=2.2$  simulates the response to luminance of the HVS (see 2.4). Since the gamma curve is steeper for low  $y(i, j)$  values, leakage errors are emphasized more than clipping errors. This approximates the behavior of the HVS, where the same luminance error is perceived as larger in leakage than in clipping.

Optimization for 2-norm has been performed on the 1.5D backlight model, at a downscaling factor of 10, thereafter PSNR has been measured in the perceptually uniform domain at low resolution. The algorithm has been compared against others presented in Section 3.4; to ensure a comparison at similar power levels, different values of the power weight factor  $q$  (1, 10 and 50) were used. The experiment was also run for  $q=0$ , but since with this setting the PSNR values were exceeding the range considered realistic, we have preferred to omit them. The results are shown in Figure 6.

As in the previous experiment, the sub-sampling strategy is capable of finding solutions that are close to the ones found by using all the samples.

5. CONCLUSIONS

In this work, we have studied a backlight dimming algorithm based on linear optimization and showed that it is possible to calculate the optimal solution using a subset of pixels varying between 9% and 26% of the total number of pixels of the original image, depending on the display model. In this way, the complexity of the problem can be reduced significantly. We have also proposed a backlight dimming algorithm based on gradient descent search. This algorithm allows to quickly improve a backlight solution obtained using any algorithm. Gradient search is particularly interesting for its versatility, especially for displays with a relatively low number of segments that allow fast computation of the gradient.

We have also considered an extended formulation to the original optimization-based algorithm based on introducing

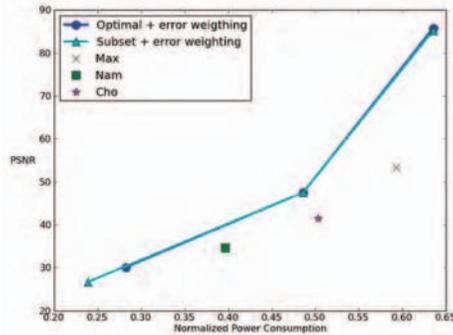


Figure 6: Effect of weighting the error perceptually during optimization on PSNR; the labels along the data lines indicate the power weight  $q$  used in the cost function.

pixel-specific error weighting and adjustable penalization of power consumption. The proposed sub-sampling strategy can find solutions that are close to optimal also when these extensions are used. All the proposed techniques can be combined together to calculate optimal backlight in a flexible and efficient manner. In the future work, we aim to improve efficiency, and explore the alternative approaches. We will also address optimization in the perceptual domain instead of physical domain. Subjective quality assessment will be performed for ultimate validation of perceived image quality.

#### REFERENCES

- [1] Shu, X., Wu, X., and Forchhammer, S., "Optimal local dimming for LED-backlit LCD displays via linear programming," Proc. SPIE-IS&T Electronic Imaging, p. 830517, San Francisco (2012).
- [2] Burini, N., Nadernejad, E., Korhonen, J., Forchhammer, S., and Wu, X., "Image dependent energy-constrained local backlight dimming," Proc. IEEE Int. Conf. Image Proc. (ICIP'12), Orlando, FL, USA (2012). Submitted for review.
- [3] Nam, H., "A color compensation algorithm to avoid color distortion in active dimming liquid crystal displays," IEEE Trans. on Consumer Electronics 56, 2569–2576 (2010).
- [4] SIM2 HDR TV, <<http://www.sim2.com/HDR/>>.
- [5] Korhonen, J., Burini, N., Forchhammer, S., and Pedersen, J. M., "Modeling LCD displays with local backlight dimming for image quality assessment," Proc. SPIE 7866, 786607–786607-9 (2011).
- [6] Albrecht, M., Karrenbauer, A., Jung, T., and Xu, C., "Sorted sector covering combined with image condensation—an efficient method for local dimming of direct-lit and edge-lit LCDs," IEICE Transactions on Electronics 93, 1556–1563 (2010).
- [7] Cho, H., and Kwon, O. K., "A backlight dimming algorithm for low power and high image quality LCD applications," IEEE Trans. on Consumer Electronics 55, 839–844 (2009).
- [8] Nam, H., "Low power active dimming liquid crystal display with high resolution backlight," Electronics Letters 47, 538 (2011).
- [9] Funamoto, T., Kobayashi, T., and Murao, T., "High-picture-quality technique for LCD televisions: LCD-AI," Proc. International Display Workshop, 1157–1158 (2001).
- [10] Kodak True Color Image Suite, <<http://r0k.us/graphics/kodak/>>.

### **A.3 Image Dependent Energy-constrained Local Backlight Dimming**

**N. Burini**, E. Nadernejad, J. Korhonen, S. Forchhammer, and X. Wu, “Image Dependent Energy-Constrained Local Backlight Dimming”, in *IEEE International Conference on Image Processing (ICIP)*, Sep. 2012, pp. 2797–2800

Reference: [5]

## IMAGE DEPENDENT ENERGY-CONSTRAINED LOCAL BACKLIGHT DIMMING

Nino Burini<sup>1</sup>, Ehsan Nadernejad<sup>1</sup>, Jari Korhonen<sup>1</sup>, Søren Forchhammer<sup>1</sup>, and Xiaolin Wu<sup>2</sup>

<sup>1</sup>DTU Fotonik, Technical University of Denmark, Kgs. Lyngby, Denmark

<sup>2</sup>McMaster University, Hamilton, Canada

### ABSTRACT

In this work, we consider and propose two extensions to an optimization-based image dependent backlight dimming algorithm. The first extension introduces error weighting based on human perception of luminance, aiming to improve the perceived image quality; the second extension adds an adjustable term for power consumption to the cost function, allowing flexible power management. Experimental results show that the proposed solution can achieve better results than other algorithms at several power consumption levels.

**Index Terms**— Liquid crystal display, Local backlight dimming, Optimization, Image quality, Power management

### 1. INTRODUCTION

Liquid Crystal Displays (LCDs) are used on a wide variety of devices, including TV sets and computer monitors. Liquid crystals do not emit light; the light is generated by the backlight instead, then modulated by the crystals, which - thanks to proper color filters - can render a large gamut. Today Light Emitting Diodes (LEDs) are common light sources for LCD backlight, due to attractive characteristics like flexibility in use and growing efficiency [1].

In LCDs a large amount of light is dissipated and only a small fraction, down to less than 10%, reaches the viewer [2]. For this reason, the backlight must emit intense light, which makes it the most power consuming component in the display. LCD are also affected by light leakage: LC cannot fully block light when reproducing black or very dark pixels, making them look slightly grayish. Light leakage is the reason for the limited contrast ratio in LCDs, and it is especially visible from wide viewing angles.

Leakage and power consumption can be reduced by dimming the backlight, and instead increase the transmittance of the LC to compensate the lower light output. This allows to decrease light emission, while rendering the same image with little or no distortion. LED technology has eased the implementation of independently controllable segments of local backlights, as opposed to global backlight, where the luminance is uniform across the whole display. In this paper, the terms "backlight segments" and LED are used interchangeably. Depending on the image content, local backlight dimming gives significant

opportunities for power saving and contrast improvement. However, aggressive dimming can cause clipping: when the LC cannot compensate the reduction of light intensity, pixels appear less bright then they should. In this work, we refer to pixels affected by clipping as "clipped" pixels.

Several backlight dimming algorithms exist; many of them focus on reducing power consumption while keeping an acceptable quality [3][4][5]. In [6], backlight dimming is modeled as an image optimization problem. The optimum backlight illumination defined by a cost function and by the input image is found; the result is a compromise between clipping and leakage. The focus is on increasing contrast and image quality and taking power savings as a positive side effect. We propose to extend this algorithm by introducing weighting of errors based on luminance perception by the Human Visual System (HVS), and by adding an adjustable term for power consumption to the cost function.

The rest of this paper is organized as follows. Section 2 presents the fundamentals of modeling LCD with backlight dimming systems. In Section 3, the proposed extensions are illustrated. Section 4 shows the experimental results and, finally, the concluding remarks are given in Section 5.

### 2. MODELING BACKLIGHT DIMMING SYSTEMS

Figure 1 illustrates a generic LCD display with local backlight dimming. The backlight dimming algorithm takes the digital image to be displayed as input, and produces the intensities for each backlight segment and the transmittance of LC pixels as output. The observed luminance  $L$  at pixel  $(i, j)$  can be calculated as the product of the backlight intensity  $B$  behind the pixel and the transmittance  $T$  of the pixels itself:

$$L(i, j) = B(i, j)T(i, j), \quad (1)$$

where  $B$  and  $T$  are normalized to the interval  $[0,1]$ .

The digital input image is typically presented in an ideally perceptually uniform format, i.e. similar intervals of pixel values represent similar differences in perceived intensity over the whole range. Since the relationship between physical luminance and perceived brightness is not linear, practical displays usually perform gamma correction when the input image is converted to LC transmittance. Gamma correction is good for conventional displays with relatively low peak luminance (up to about 100 cd/m<sup>2</sup>), but more accurate functions have been proposed for perceptual linearization of the input signal in brighter displays [7].

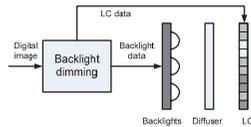


Figure 1. Generic local backlight dimming system.

Since HVS is very sensitive to contrast changes at low physical luminance levels, light leakage becomes a problem in dark parts of images. The impact of leakage on the observed luminance can be included by modifying Eq. (1):

$$L(i, j) = B(i, j)T(i, j) + \epsilon(i, j)B(i, j)(1 - T(i, j)), \quad (2)$$

where  $\epsilon(i, j)$  is a leakage factor representing the amount of light leaking through a black LC pixel. Leakage and  $\epsilon(i, j)$  depend on pixel position and viewing angle; however, in this work, we assume a constant  $\epsilon$  across the display.

In practical LCDs, light from independent backlight segments is diffused and mixed on the diffuser plate between backlight units and LC layer. In order to compute the backlight intensity at each pixel, we need to know how the light is distributed around the center of a light source. It is assumed that the distribution follows a Point Spread Function (PSF) specific to the display, multiplied by the normalized luminance of the backlight unit. Each pixel position is usually influenced by several backlight units, and the total luminance can be computed simply by summing the individual contributions. The PSF is sometimes approximated by a two-dimensional Gaussian function, but for more accurate results, the PSF should be determined experimentally for each display. The backlight intensity at pixel  $(i, j)$  can be computed as follows:

$$B(i, j) = \sum_{k=1, N} B_k h_k(i, j), \quad (3)$$

where  $B_k$  is the intensity of backlight unit  $k$ , and  $h_k$  is the PSF expressing the relative amount of light from backlight  $k$  reaching pixel  $(i, j)$ . Since the backlight dimming algorithm may have dimmed the light intensity significantly in the regions dominated by dark pixels, the so-called "brightness compensation" is performed to restore the original target luminance. This is done by increasing the LC transmittance  $T$  so that the target  $L$  is reached.

Using the Eqs. (2) and (3), it is possible to compute the observed luminance of each pixel, or in case of color display, the luminance of R, G and B subpixels, out of the intensities of individual backlights and LC transmittance values. The modeled luminance can be converted back to uniform scale by performing inverse Gamma correction and rescaling normalized values to 8-bit integers. Then, it is possible to compare the modeled image on the display against the original input using traditional measures, such as

Mean Squared Error (MSE), Peak Signal-to-Noise Ratio (PSNR), or any other image quality indicator.

### 3. PROPOSED ALGORITHM

An algorithm minimizing the error was presented in [6]. In this work, we propose two extensions: one extension is image dependent individual weighting of the pixel error, based on perceptual aspects; the other extension is the introduction of a term representing power consumption in the cost function.

As in [6], the algorithm models backlight dimming as an optimization problem. The distortion to be minimized is the difference between an ideal output  $y$  and the actual output  $x$  rendered on the screen, which is determined by the backlight  $b$  as derived from Eq. (3), the compensated pixel values  $a$  and the leakage factor  $\epsilon$ ; the constraints include the values of pixel transmittance and LED intensity. The other term is power  $p$  multiplied by a weighting parameter  $q$ . Optimization can be done in  $\ell_1$  or  $\ell_2$ ; the experiments in this work used  $\ell_1$  as it allows to solve the problem using linear programming. For more details, the reader may consult [6]. The formulation of the problem, with the proposed extensions, is given as:

$$\begin{aligned} & \text{minimize } \|(y - x) \circ w\| + q \times p \\ & \text{subject to } x = a \circ b \\ & \quad b = Hr \\ & \quad \epsilon \leq a \leq 1 \\ & \quad 0 \leq r \leq 1 \end{aligned} \quad (4)$$

where the constraint  $b = Hr$  corresponds to Eq. (3) and the term  $(y - x)$  is multiplied point-wise by a weighting matrix  $w$ . The original formulation of the optimization problem [6] can be obtained again by setting  $w$  to a matrix of ones and by setting  $q$  to zero.

The original formulation solves the backlight problem in the physical domain, meaning that it deals with luminance and pixel transmittance to find the light output that matches the target as closely as possible. However, the HVS has a nonlinear response to luminance, and the perceived luminance  $L_p$  is a function of the physical luminance  $L$ :  $L_p = f(L)$ . The function  $f$  can be modeled in several ways. Without loss of generality, we can approximate the HVS response to luminance with an inverse gamma function:

$$f(L) = L^{1/\gamma} \quad (5)$$

A typical value for  $\gamma$  is 2.2 [8]. Due to the nonlinearity of HVS, a small change of a low luminance level will be more noticeable than the same change at a higher luminance level. This implies that, given the same absolute error in physical luminance, leakage is more noticeable than clipping. For this reason, the optimal "physical" solution to the backlight problem might not be the optimal perceived solution. We propose to modify the cost function by adding a weighting matrix  $w$  that assigns a weight to the error, depending on the target luminance of the pixel in question.

$$w_{i,j} = f'(y_{i,j}) \quad (6)$$

## A.3 Image Dependent Energy-constrained Local Backlight Dimming<sup>17</sup>

In this work, the weight is calculated from the gamma curve at the normalized target luminance value. Since the slope of the curve is steeper at low luminance, errors in the dark pixels will be weighted more than those in the high luminance area, providing a better match to the perception model. The concept of the weighting matrix  $w$  can be extended to include any kind of image dependent weighting applied to the error which is to be minimized.

The second extension proposed adds a power term to the cost function. The optimization-based algorithm aims to achieve the best possible image quality and considers power savings as a positive side effect. However, lower image quality might be acceptable in exchange of lower power consumption, and it is interesting to study the tradeoff between the two factors. The power term in the cost function includes the mean of the normalized LED intensities (Eq. (7)), which estimates power consumption:

$$p = \frac{\sum_{i,j} B_{ij}}{N} \quad (7)$$

### 4. EXPERIMENTAL RESULTS

The experiments conducted here aim at evaluating the performance of the proposed algorithm with different power weights in the cost function and at different image resolutions. The first experiment compares the impact of image downscaling for the proposed algorithm on two simulated displays. The second experiment compares the proposed algorithm against other algorithms as the power weight factor and the image downscaling factor vary.

The proposed algorithm has been compared with other algorithms described below. In the Max and Avg algorithm [9], the intensity of each backlight segment is set to the maximum or average value of the segment pixels; only the Max algorithm has been considered for the experiment. The other algorithms cited here are referred to by the name of the first author of the respective publication. In Cho et al. [3], the algorithm analyzes each segment and sets the backlight luminance to the sum of the average value of the segment pixels and a correction value, which depends on the average and the maximum value of the segment. In Nam [4], the backlight luminance can be set locally or globally depending on the type of image. LED intensities for global and local cases are set according to the mean of the maximum and average values of the total image and segment, respectively. Finally, in Albrecht et al. [5] a clipper-free algorithm consisting of three steps is proposed: the first step sets lower bounds for each backlight segment, depending on the image content and on the PSF; the optional second step is iterative: for each iteration, it increases the luminance of one LED until there are no clipped pixels, or another ending condition is met; the third step scans the pixels of each segment in a specific order and adjusts the LED values to make sure that every pixel receives enough backlight.

Two screen models have been used for the experiments. The first one is a simulated device with 221 backlight segments placed in a grid of 13 rows and 17 columns; the

PSF is given by a Gaussian function. The second one is modeled on a 47" local backlight dimming Full HD screen manufactured by SIM2 [10]. The screen has 2202 backlight segments placed in a hexagonal grid behind the LC layer; the PSF has been provided by the manufacturer. For our simulations, leakage has been modeled with an  $\epsilon$  value of 0.001 for both screens.

Eight images were used in the experiments. The set of chosen images includes several image types like synthetic graphics, natural images and compressed pictures. For the simplicity of error analysis, all the images are in grayscale.

The target luminance  $y$  is calculated from the sRGB input image  $I$  as the CIE 1931 luminance component  $Y$  (8). The complexity of the proposed algorithm grows very quickly with the image size; e.g., solving the optimization problem for a Full HD image on the SIM2 screen implies millions of variables and constraints. It is possible to reduce the complexity by downscaling the input images. Depending on the experiment, the input images have been downscaled by a factor varying between 1 (original size) and 10 using bicubic interpolation. The PSF of the screens have been downscaled accordingly. Even if the backlight is calculated from a downscaled version of the input image, the resulting  $B_{ij}$  values are applied at full resolution; after brightness compensation, distortion is measured in PSNR between the normalized input image  $I$  and the output of the algorithm  $x$  after applying Eq. (5) to each  $L(i,j)$  obtained from Eq. (2) at full resolution:

$$PSNR = 10 + \log_{10} \left( \frac{N}{\sum_{i,j} (I(i,j) - x(i,j))^2} \right) \quad (8)$$

where  $N$  is the number of pixels in  $I$  and  $x$ . Figure 2 shows that both the PSNR and power consumption of the solution obtained from the proposed algorithm increase as the downscaling factor decreases; the factor was varied between 1 and 10 for the screen with Gaussian PSF, and between 4 and 10 for the SIM2 screen.

Figure 3 shows the comparison between the proposed method and the other algorithms presented earlier. The results have been calculated for the optimal backlight with 13 positive values of the power weight factor  $q$  (Eq. 4). The resulting curve highlights the tradeoff between power consumption and PSNR. For a better power level, the proposed solution can achieve a given outcome than the other algorithms: the PSNR for the proposed algorithm is approximately 3db higher than the Max, Nam, Cho and Albrecht solutions at the power level of each of these; with the same PSNR, the proposed algorithm reduces power consumption by 0.2 compared to Albrecht. It is interesting to notice that for low  $q$  values (right part of the curve in the figure) a modest increase of error is associated with a significant reduction in power consumption.

Table 1 shows the relative contributions to MSE from leaking and clipped pixels for all algorithms. Three  $q$  values are considered for the proposed method. Using the notation of problem (4), pixel  $(i,j)$  is defined as leaking if  $a = 0$  and  $x(i,j) > y(i,j)$  and it is defined as clipped if  $a = 1$  and

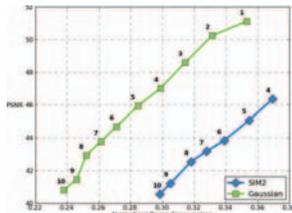


Figure 2. Impact of downscaling factor (above the dots) on PSNR and power consumption.

$x(i,j) < y(i,j)$ . The error in Max, Nam and Cho is almost completely caused by clipping, as they fail to provide enough light to many pixels. The error in the Albrecht algorithm is mostly due to leakage instead; this is expected, since the algorithm is clipper free. There is a small clipping error caused by the algorithm being run on a downscaled image, causing clipping in the full resolution image. As for the proposed algorithm, the impact of leakage and clipping depends on the power factor  $q$ . Leakage is dominant when  $q$  is equal to zero, but clipping pixels become more important as  $q$  increases. This is easily explained by the fact that a higher  $q$  favors low LED intensities, resulting in more pixels being clipped.

5. CONCLUSIONS

We have presented two extensions for an optimization-based image dependent backlight dimming algorithm [6] to provide a tradeoff between perceived distortion and power. The proposed solution can achieve a PSNR improvement of about 3db compared to other algorithms at the level of power consumed by these. Downscaling the input image prior to optimization reduces complexity and power consumption, although at the price of a higher mean square error. Leakage is the main cause for errors at high power levels, while clipping becomes dominant at low levels. Future work will include reducing the complexity of the algorithm, and investigation of the perceptual weighting of errors.

Table 1: average absolute contribution of leaking and clipped pixels on MSE ( $\cdot 10^{-3}$ ) and related percentage.

Algorithm	Leakage	Clipping
Max	2.49	4.92%
Nam	0.84	0.42%
Cho	1.91	2.19%
Albrecht	25.1	97.73%
Proposed $q=0$	5.5	65.50%
Proposed $q=1468$	2.99	17.61%
Proposed $q=19818$	0.88	0.66%

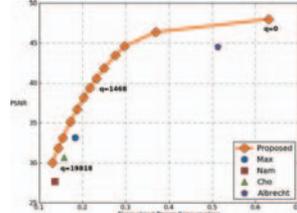


Figure 3. Comparison of the proposed solution against other algorithms at various power weights  $q$  (Eq. 4).

6. REFERENCES

- [1] M. Anandan, "Progress of LED backlights for LCDs," *Journal of the SID*, vol. 16, no. 2, pp. 287-310, 2008.
- [2] H. Groot Hulze and P. de Greef, "Power Savings by Local Dimming on a LCD Panel with Side Lit Backlight," *SID Symposium Digest of Technical Papers*, vol. 40, no. 1, pp. 749-752, 2009.
- [3] H. Cho and O. K. Kwon, "A backlight dimming algorithm for low power and high image quality LCD applications," *Consumer Electronics, IEEE Transactions on*, vol. 55, no. 2, pp. 839-844, 2009.
- [4] H. Nam, "Low power active dimming liquid crystal display with high resolution backlight," *Electronics Letters*, vol. 47, no. 9, p. 538, 2011.
- [5] M. Albrecht, A. Karrenbauer, T. Jung, and C. Xu, "Sorted Sector Covering Combined with Image Condensation—An Efficient Method for Local Dimming of Direct-Lit and Edge-Lit LCDs," *IEICE Transactions on Electronics*, vol. 93, no. 11, pp. 1556-1563, 2010.
- [6] X. Shu, X. Wu, and S. Forchhammer, "Optimal local dimming for LED-backlit LCD displays via linear programming," in *Proceedings of SPIE-IS&T Electronic Imaging*, 2012, p. 830517.
- [7] J. Korhonen, N. Burini, S. Forchhammer, and J. M. Pedersen, "Modeling LCD displays with local backlight dimming for image quality assessment," in *Proceedings of SPIE-IS&T Electronic Imaging*, 2011, vol. 7866, pp. 786607-786607-9.
- [8] M. Stokes, M. Anderson, S. Chandrasekar, and R. Motta, "A Standard Default Color Space for the Internet - sRGB," *Microsoft and Hewlett*, pp. 238-245, 1996.
- [9] Funamoto T, Kobayashi T, and Murao T, "High-Picture-Quality Technique for LCD televisions: LCD-AI," in *Proc Int Disp Workshops*, 2001, pp. 1157-1158.
- [10] <http://www.sim2.com/HDR/>

## A.4 Enhancing Perceived Quality of Compressed Images and Video With Anisotropic and Fuzzy Filtering

E. Nadernejad, J. Korhonen, S. Forchhammer, and **N. Burini**, “Enhancing Perceived Quality of Compressed Images and Video with Anisotropic Diffusion and Fuzzy Filtering”, *Signal Processing: Image Communication*, vol. 28, no. 3, pp. 222–240, Mar. 2013

Reference: [7]



## Enhancing perceived quality of compressed images and video with anisotropic diffusion and fuzzy filtering

Ehsan Nadernejad, Jari Korhonen\*, Søren Forchhammer, Nino Burini

Department of Photonics Engineering, Technical University of Denmark (DTU), Kgs. Lyngby, Denmark

### ARTICLE INFO

**Article history:**  
Received 23 December 2011  
Accepted 4 December 2012

**Keywords:**  
Fuzzy filter  
Anisotropic diffusion  
H.264/AVC  
Visual quality

### ABSTRACT

Fuzzy filtering has recently been applied and optimized for reducing distortion in compressed images and video. In this paper, we present a method combining the powerful anisotropic diffusion equations with fuzzy filtering for removing blocking and ringing artifacts. Due to the directional nature of these artifacts, we have applied directional anisotropic diffusion. In order to improve the performance of the algorithm, we select the threshold parameter for the diffusion coefficient adaptively. Two different methods based on this approach are presented: one designed for still images and the other for YUV video sequences. For the video sequences, different filters are applied to luminance (Y) and chrominance (U,V) components. The performance of the proposed method has been compared against several other methods by using different objective quality metrics and a subjective comparison study. Both objective and subjective results on JPEG compressed images, as well as MJPEG and H.264/AVC compressed video, indicate that the proposed algorithms employing directional and spatial fuzzy filters achieve better artifact reduction than other methods. In particular, robust improvements with H.264/AVC video have been gained with several different content types.

© 2012 Elsevier B.V. All rights reserved.

### 1. Introduction

Image and video compression is a common source of spatial and temporal distortion. Spatial distortion includes blocking and ringing, whereas typical temporal distortion types are mosquito and flickering artifacts. Blocking artifacts are caused by separate compression of each block, and it occurs both in horizontal and vertical direction of each frame. Ringing artifacts occur when the high frequency transform coefficients obtained from discrete cosine transform (DCT) or wavelet-based coding are quantized or truncated. This causes ripples or oscillations around sharp edges or contours in the image, known as Gibbs phenomenon. When ringing artifacts alternate from

frame to frame as a video sequence is displayed, mosquito artifacts are created. Flickering artifacts [1] appear due to the quality inconsistencies at the same spatial position in adjacent frames.

Even though blocking and ringing artifacts do not necessarily cause significant distortion in terms of measurable noise, the human visual system (HVS) has shown to be relatively sensitive to such artifacts. Over the years, many algorithms have been proposed to reduce the spatial and temporal artifacts. Zhai proposed an algorithm for deblocking [2], consisting of three parts: local AC coefficient regularization (ACR) of shifted blocks in the discrete cosine transform (DCT) domain, block-wise shape adaptive filtering (BSAF) in the spatial domain, and a quantization constraint (QC) in the DCT domain [2]. Kim [3] proposed an adaptive deblocking algorithm for low bit-rate video coding. In that algorithm, the DC and AC values of each block are used to classify each block into

\* Corresponding author. Tel.: +4545256594.  
E-mail address: jari@fotonik.dtu.dk (J. Korhonen).

0923-5965/\$ - see front matter © 2012 Elsevier B.V. All rights reserved.  
<http://dx.doi.org/10.1016/j.image.2012.12.001>

Please cite this article as: E. Nadernejad, et al., Enhancing perceived quality of compressed images and video with anisotropic diffusion and fuzzy filtering, Signal Processing-Image Communication (2012), <http://dx.doi.org/10.1016/j.image.2012.12.001>

## ARTICLE IN PRESS

2

E. Nadernejad et al. / Signal Processing: Image Communication 1 (2012) 100–108

one of two categories, low and high activity block. In the following post-processing stage, two kinds of low-pass filters are adaptively applied on each block, based on the classification result. In [4,5], linear low-pass filters, and Projection Onto Convex Sets (POCS) in [6] have been used for post-processing. Anisotropic diffusion has been proposed to remove blocking artifacts, exploiting the characteristic of HVS [7]. A de-blocking algorithm [8] has been proposed for DCT-based compressed images, using anisotropic diffusion that can control the diffusion rate along the direction of the edges, using a rate control parameter. Yao et al. [9] proposed a post-processing algorithm for reducing coding artifacts in compressed image and video sequence. The algorithm is based on anisotropic diffusion process, using a histogram-driven diffusion coefficient.

The methods listed above can reduce blocking artifacts, but they introduce blurriness. To remove ringing artifacts, linear or non-linear isotropic filters can be applied to the regions near to edges [10,11]. For combating flickering artifacts, most of the current methods focus on reducing the flickering in intra-frame coding [1,12], where flickering artifacts are often especially visible. In [1], the quantization error is considered for deciding the optimal intra prediction mode to reduce flickering. Also in [12], flickering is included in the cost function, when the optimal prediction mode and block size are chosen. In [11,13–15], spatiotemporal fuzzy filters are used to remove different artifacts. Other methods for artifact removal have been introduced in [16–21].

It is well known that orientation and frequency are the fundamental spatial characteristics processed by the HVS [22,23]. In order to produce visually pleasing results, these characteristics need to be taken into account in image enhancement algorithms. It has been observed that the use of Gaussian-like filtering for removal of small-scale spatial artifacts is motivated from the perspective of human perception [24]. In image processing and computer vision, anisotropic diffusion, also called *Perona-Malik* diffusion, is a technique aiming at reducing the noise without removing essential parts of the image content, such as edges, lines and other details that are important for the interpretation of the image. Their use for image restoration and enhancement has been studied extensively [25–40]. In particular, Ling and Bovik [38] introduced smoothing of low signal-to-noise ratio (SNR) medical images via regularization of the anisotropic diffusion using median filtering.

Another type of filter used in our study is the fuzzy filter, which is derived from the fuzzy transformation theory [41], and has been applied to coding artifacts reduction recently [11–14]. This filtering technique directed by the classified edge map provides a solution for coding artifacts reduction, but the perceptual quality of the processed images and videos is however not optimal with respect to the blocking and ringing artifacts. On the other hand, anisotropic diffusion is a powerful algorithm for noise reduction and image enhancement and it may reduce coding noise in general.

In this paper, we present two new methods based on combining the powers of anisotropic diffusion processing and spatial fuzzy filtering [13,14] to reduce coding

artifacts in compressed images and video, thus providing an improved solution for the artifact reduction and perceptual quality enhancement. We process all the vertical and horizontal artifacts using one-dimensional (1D) anisotropic diffusion after applying a fuzzy filter. To avoid the blurring effect, anisotropic diffusion is performed with a small number of iterations. This paper is an extended version of our initial work [42], including a more comprehensive description of the extended algorithm and a significantly expanded results section, including a subjective comparison study to verify the results obtained with traditional objective quality measures. Additional algorithms have also been included for comparison.

Our results show that the proposed methods give good results with both still images and video sequences, even for H.264/AVC compressed video, that has not been widely covered in studies for video denoising algorithms up to date. Since H.264/AVC represents the state of the art in video compression, we have compared the performance of the proposed method against methods proposed also in [2–5,11,13,14,16,17,43] for both H.264/AVC compressed video sequences, and JPEG coded images. In [13], results using a spatiotemporal approach for H.264/AVC encoded video are also presented.

The rest of the paper is organized as follows: Section 2 provides a description of fuzzy filtering. Section 3 summarizes the anisotropic diffusion. Combined adaptive fuzzy filtering and anisotropic diffusion for artifact reduction in compressed video sequences is explained in Section 4. Section 5 summarizes the techniques used for performance evaluation. Section 6 shows the experimental results and compares the proposed algorithm with known methods in terms of visual quality measured both objectively and subjectively. Finally, the concluding remarks are given in Section 7.

## 2. Fuzzy filter

Fuzzy filtering has been used with success in deblocking and deringing of compressed images and videos [11,13,14,41]. Fuzzy filters, such as those described in [13,14], are designed for similar purposes as median filters [15] or rank condition rank selection filters [41]. In this section, we describe the fuzzy filter following [13,14]. Assuming that a given filter  $h$  is applied to a set  $\omega$  of neighboring pixels around the input pixel at  $(i, j) \in Z^2$ , we can formulate the normalized output:

$$\tilde{f}(i, j) = \frac{\sum_{(i', j') \in \omega} h(i' + i, j' + j) f(i', j')}{\sum_{(i', j') \in \omega} h(i' + i, j' + j)}, \quad (1)$$

where  $h(f(i' + i, j' + j)) f(i', j')$  controls the weight of the input pixel intensities  $f(i' + i, j' + j)$ .

A low-pass filter designed to perform effectively in the flat areas may introduce blurring artifacts in the detailed areas [14]. The challenge is to preserve the details, while removing the artifacts.

The function  $h(f(i' + i, j' + j), f(i', j'))$  is called a membership function [11,15,41]. In our work, a Gaussian membership

function is used as given in [13,14]:

$$h(\lfloor i+\tilde{i}, j+\tilde{j} \rfloor, \lfloor i, j \rfloor) = \exp\left(-\frac{(\lfloor i+\tilde{i}, j+\tilde{j} \rfloor - \lfloor i, j \rfloor)^2}{2\sigma^2}\right), \quad (2)$$

where  $\sigma$  represents the spread parameter of the input and depending on  $i, \tilde{i}, j, \tilde{j}$  and  $f$  (Fig. 3) controls the strength of the fuzzy filter. The input  $\lfloor i, j \rfloor$  always contributes no less to the output than the other samples:

$$h(\lfloor i, j \rfloor, \lfloor i, j \rfloor) \geq h(\lfloor i+\tilde{i}, j+\tilde{j} \rfloor, \lfloor i, j \rfloor). \quad (3)$$

For the same difference  $\|\lfloor i+\tilde{i}, j+\tilde{j} \rfloor - \lfloor i, j \rfloor\|$ , the higher the value of  $\sigma$ , the higher the contribution of  $\lfloor i+\tilde{i}, j+\tilde{j} \rfloor$  to the output. This implies that  $\lfloor i, j \rfloor$  will be influenced more by  $\lfloor i+\tilde{i}, j+\tilde{j} \rfloor$ . Smaller values of  $\sigma$  will keep the signal  $\lfloor i, j \rfloor$  more isolated from its neighboring samples. The spread parameter should be adaptive to different areas with different activity levels, such as smooth or detailed textures. The conventional fuzzy filter uses a fixed spread parameter in (2) for the surrounding sample, ignoring their relative positions. Adaptive influence can be achieved by an adaptive spread parameter:

$$\sigma(\lfloor i+\tilde{i}, j+\tilde{j} \rfloor, \lfloor i, j \rfloor) = K(\lfloor \tilde{i}, \tilde{j} \rfloor) \times \sigma_a(\lfloor i, j \rfloor), \quad (4)$$

where  $\sigma_a$  is a position-dependent amplitude function of the spread parameter, and  $K$  is the scaling function controlled by the direction of  $\lfloor i+\tilde{i}, j+\tilde{j} \rfloor$  to  $\lfloor i, j \rfloor$ . We use  $K=1$ , as in [8]. In this work, we use 1D fuzzy filtering for deblocking and 2D fuzzy filtering for deringing. 1D fuzzy filter can be derived from the definitions for 2D filtering above by removing one of the dimensions from Eqs. (1)–(4).

In image and video compression, distortions such as blocking, ringing or flickering artifacts are directional; thus, the direction between  $\lfloor i, j \rfloor$  and its surrounding samples  $\lfloor i+\tilde{i}, j+\tilde{j} \rfloor$  should be taken into consideration. In [14], a cosine-based general form for the spread parameter was used:

$$\sigma(\theta) = \sigma_a(\alpha + \beta \cos^2(\theta)), \quad (5)$$

where  $\sigma_a$  is a function of the standard deviation of pixel intensities as described in [8],  $\theta$  is the angle defining the direction from the pixel of interest  $\lfloor i, j \rfloor$  to the surrounding pixel  $\lfloor i+\tilde{i}, j+\tilde{j} \rfloor$ , relative to the horizontal direction,  $\alpha$  and  $\beta$  are positive scaling factors controlling the maximum and minimum strength of the filter [9]. The details of the membership function  $\sigma$  in Eq. (3) are discussed in [13,14] for compressed images and compressed video sequences. More information about fuzzy filters is available in [11,15,41].

### 3. Anisotropic diffusion equations

Anisotropic diffusion is a mathematical model with many practical applications in physics and chemistry. The use of anisotropic diffusion has been widely extended for signal and image processing [25–40]. This method was first proposed by Perona and Malik [25] for multiscale description, enhancement, and segmentation of images. Let a gray scale and 2-D image  $I(x, y)$  be represented by a real-valued mapping  $I: \mathbb{R}^2 \Rightarrow \mathbb{R}$ . In P-M diffusion, the initial image  $I_0$  is modified through the anisotropic

diffusion equation:

$$\frac{\partial I}{\partial t} = \text{div}[c(\|\nabla I\|)\nabla I] \\ I(x, y, t) \Big|_{t=0} = I_0(x, y), \quad t \in (0, T), \quad (6)$$

where  $\text{div}$  denotes the divergence operator,  $\nabla$  is the gradient operator,  $\|\nabla I\|$  is the gradient magnitude of  $I$ ,  $I(x, y, t)$  is the two-dimensional image as a function of position  $(x, y)$  at time  $t$ , where  $t$  is a scale parameter; in discrete implementation it is used to enumerate iteration steps (step size) [28], and  $c(\|\nabla I\|)$  is a diffusion coefficient [25–30] which controls the diffusion speed. This function is a monotonically decreasing function of the gradient magnitude  $c(\|\nabla I\|)$ . It yields intra-region smoothing but not inter-region smoothing [35–39] by impeding the diffusion at image edges. It increases smoothing parallel to the edge and stops smoothing perpendicular to the edge, as the highest gradient values are perpendicular to the edge and dilated across edges. The choice of  $c(\|\nabla I\|)$  can greatly affect the extent to which discontinuities are preserved. If  $c(\|\nabla I\|)$  is allowed to vary according to the local image gradient, then we have anisotropic diffusion. Two different diffusion coefficients were proposed by PM. In this paper, we use  $c(\|\nabla I\|)$  as in [25]:

$$c(x, y, t) = \exp\left(-\frac{\|\nabla I(x, y, t)\|^2}{k^2}\right), \quad (7)$$

where  $k$  is a threshold parameter used to distinguish edge region from smooth region.

The major drawback of the above-mentioned method is that the gradient is computed from the noisy image. From a practical point of view, the place of edges in the image may not be recognized accurately [26,28]. Theoretically it is ill-posed in the sense that similar images are likely to diverge during the diffusion process [35]. In order to overcome this problem, it was suggested to use a regularized or smoothed version of the image to calculate the gradient [28]. In our approach, the gradient is calculated from a smoothed image [29,30]:

$$\nabla(I_G) = \nabla(G_\sigma * I), \quad (8)$$

where  $G_\sigma$  is a Gaussian filter, and  $*$  is a convolution operator. Instead of Gaussian filtering, Ling [38] and Demirkaya [32] introduced another version of regularization by median filtering to obtain better edge estimation. In this case, in the anisotropic diffusion equation in Eq. (6), the diffusion function is replaced by  $\nabla(I_M)$ , where  $I_M$  is the image  $I(x, y, t)$  after median filtering. In addition, in [39] a relaxed median filter in the diffusion steps is used to remove noise from the image. The blocking effect is the main problem of median filtering in anisotropic diffusion equation, and this is why Gaussian regularization is used to implement the anisotropic diffusion equation for deringing (and deblocking).

Parameter  $k$  in the diffusion function in Eq. (7) has been chosen as suggested in [42]. In our work, it will be used for the horizontal and vertical directions. The value is derived based on the gradient of the image:

$$k = \varphi \cdot B, \quad (9)$$

Please cite this article as: E. Nadernejad, et al., Enhancing perceived quality of compressed images and video with anisotropic diffusion and fuzzy filtering, Signal Processing-Image Communication (2012), <http://dx.doi.org/10.1016/j.image.2012.12.001>

where  $\phi$  is a constant and  $B$  is the variance of the image gradient in different directions. The value of  $B$  is computed using the variance of the gradient magnitude of the image as follows:

$$B = \text{var}(|\nabla I|). \quad (10)$$

This will make the diffusion function adaptive according to the gradient changes of the image in different directions. We chose  $\phi = 2 \times 10^{-6}$ , as suggested in [42]. The anisotropic diffusion equation in Eq. (3) is solved using the finite difference method [35–37]. The spatial derivatives are approximated by central differences, while the temporal derivative is replaced by its forward difference approximation. This results in an explicit finite-difference equation of the form:

$$I_{ij}^{n+1} = I_{ij}^n + \frac{\Delta t}{|\eta_{ij}|} \sum_{p \in \eta_{ij}} c(|\nabla I_{ij}^n|_p) \cdot c(|\nabla I_{ij}^n|_p), \quad (11)$$

where  $I_{ij}^n$  is the discretely sampled image with pixel position  $(i, j)$ ,  $\eta_{ij}$  is the spatial neighborhood of the pixel  $(i, j)$ ,  $|\eta_{ij}|$  is the number of pixels in this neighborhood window,  $c(\cdot)$  is diffusion coefficient given by Eq. (7),  $n$  is the number of iteration and  $\Delta t$  is the time step size. An approximate solution to this equation can be obtained in the discrete domain [30]. The spatial discretization used in Section 4.1.3 applies to  $1 \times 3$  or  $3 \times 1$  pixel windows in the vertical and horizontal directions, and corresponds to the simpler standard discretization and the more complex non-negativity discretization [33,37]. The one-dimensional discrete form, which is used in the proposed method, will be explained in Section 4.1.3.

The use of diffusion equations has already been established as an important method for image denoising. A number of authors have addressed color image enhancement using diffusion equations [36,44–47]. Tang et al. [36] introduced an algorithm to remove noise from multi-channel images. The algorithm is based on separating the image data into chromaticity and brightness components, and then processing the components with partial differential equations or diffusion flows. In Tang's algorithm, each color pixel is considered as an  $n$ -dimensional vector. The two chroma components are processed using cross component diffusion equations.

When RGB images are considered, image enhancement performed separately on each color channel may result in smearing or blurring of edges that are not present in all the channels. Prasath [44] proposed a multichannel version of partial differential equation, which is used to restore noisy color images. Weighted coupling of inter-channel edges is done by computing the Laplacian differences to detect edges between channels. Anisotropic intra-channel smoothing is then used to denoise and preserve edges. Weickert [45] proposed a method for enhancing coherent flow-like structures in color images, based on anisotropic diffusion. To avoid the enhancement process to evolve in different directions in the R, G and B channels, a common diffusion tensor is used to all the channels. Sapiro et al. [46] perform image diffusion via coupled differential equations in a framework generalized to any color space, such as CIE  $L^*a^*b^*$ . The algorithms in [45,46] use a system of single-valued images (channels),

scalar PDEs, each performed on a separate color channel. Boccignone et al. [47] proposed another approach, using the color channel interactions in the framework of the thermodynamics of open systems. It has been shown that by considering different channels as interacting systems, it is possible to derive a generalized diffusion equation that determine the evolution in the spatio-chromatic scale space without being constrained by a particular form of diffusivity. We focus on video frames in YUV format. Here the luminance and chrominance have different statistical characteristics and perceptual importance. Furthermore, YUV values are less correlated than RGB values. Therefore we take the computationally simpler approach of processing Y, U, and V separately. Cross-channel processing may be beneficial for removing e.g. color bleeding artifacts.

#### 4. Proposed method for removing artifacts

Fuzzy filtering has been proven efficient in deblocking and deringing [11,13–15], whereas anisotropic diffusion is commonly used for image enhancement [26–31]. Since compressed images and video frames in real life typically suffer from noise artifacts in addition to blocking and ringing, we consider a combination of fuzzy filtering and anisotropic diffusion to be a reasonable approach for general image enhancement. In this section, we explain how the adaptive anisotropic diffusion and fuzzy filtering are combined for artifact reduction in compressed images and video sequences (MPEG, H.264/AVC). We define two different methods based on the filters described in Sections 2 and 3: The first proposed method is designed for still images, and the second method for video sequences. The main difference between these methods is that for video sequences, luma and chroma components are handled differently, as they perceptually have different significance, and they are also separated in the native video format (YUV).

##### 4.1. Spatial fuzzy filtering and directional anisotropic diffusion for deblocking and deringing

In the proposed method, adaptive 1D fuzzy filtering is first applied to the pixels potentially suffering from blocking artifacts, and then directional anisotropic diffusion is used to increase the quality. The 1D fuzzy filter follows Eqs. (1)–(4) and is similar to [11], except that we use a Gaussian kernel function as defined in Section 3 instead of a linear function. Finally, adaptive 2D fuzzy filtering [13] is applied to the pixels with ringing artifacts. An overview of the combined method is given in Fig. 1.

When adaptive fuzzy filtering is used, the blocking artifacts will be attenuated or removed from the image, but the unwanted noise remains, if 1D adaptive anisotropic diffusion is removed (in both directions) from the system depicted in Fig. 1. This is illustrated in Fig. 2b: The noise remains (see the background of the image), but the blocking effect is removed (compare the face of the kid and the mother in Fig. 2a and b). On the other hand, if anisotropic diffusion is used without 1D fuzzy filtering, the quality of the image is improved by removing noise

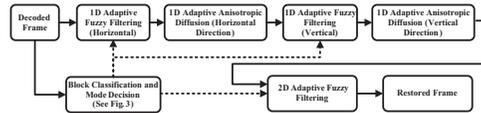


Fig. 1. Flowchart of the proposed method for image filtering. Solid lines denote the data flow and dashed lines the parameter flow.



Fig. 2. The effect of removing the adaptive fuzzy filter and adaptive anisotropic diffusion: (a) the original compressed image, (b) 1D adaptive fuzzy filtering (both directions) and (c) 1D adaptive anisotropic diffusion (both directions).

(in the background), but some blocking artifacts will remain. Fig. 2c shows the effect of the anisotropic diffusion: The quality is improved, but blocking effect remains in several places.

4.1.1. Block classification

At this step, we classify the blocks to allow 1D and 2D fuzzy filter to adapt to the variation of the pixels in each block. This is based on standard deviation (STD) of  $3 \times 3$  pixel blocks, denoted as  $STD(I_{ij})$  when the center pixel of the block is at position  $[ij]$ . The classification is based on comparing the STD value in a  $3 \times 3$  windows around each pixel and the value of the maximum STD (MaxSTD) in a  $4 \times 4$  block (with H.264/AVC) or an  $8 \times 8$  block (with JPEG/MJPEG) with a set of predetermined threshold. The threshold values for different labels are obtained by repeating the experiments on JPEG compressed images, deblocking and deringing them using different thresholds, and choosing the values giving the best results.

$$Label = \begin{cases} \text{strongedgeblock} & \text{if } \text{MaxStd} \in [45, +\infty) \\ \text{weakedgeblock} & \text{if } \text{MaxStd} \in [25, 45) \\ \text{strongtextureblock} & \text{if } \text{MaxStd} \in [15, 25) \\ \text{weakTextureblock} & \text{if } \text{MaxStd} \in [5, 15) \\ \text{smoothblock} & \text{if } \text{MaxStd} \in [0, 5) \end{cases} \quad (12)$$

Fig. 3 shows the flowchart of the proposed algorithm to find the spread parameter ( $\sigma$ ) for a block. The main differences between the proposed method and the method in [11] are that a linear membership function is used as a kernel function instead of Gaussian function, and images are divided to blocks of  $4 \times 4$  (for H.264) or  $8 \times 8$  pixel blocks, and then classified in five categories.

In contrast, four categories for  $8 \times 8$  pixel blocks are used in [11].

4.1.2. One-dimensional adaptive fuzzy filtering

In order to remove blocking artifacts in horizontal and vertical directions, an adaptive fuzzy filtering will be applied in each direction, as described in Section 2. Vertical artifact detection is performed along each vertical boundary of an  $8 \times 8$  or  $4 \times 4$  block, which are the relevant sizes for most of the practical block-based codecs employing transform of  $8 \times 8$  (as JPEG and MJPEG) or  $4 \times 4$  pixels (as H.264/AVC). First, the difference between each pair of boundary pixels,  $G_0$ , is found (see Fig. 4);  $G_0$  is the absolute difference between  $Y_1$  and  $X_0$  ( $8 \times 8$  pixels) or  $Y_2$  and  $X_0$  ( $4 \times 4$  pixels). The absolute differences between pixels are denoted using  $R$  and  $L$ , right and left of the border, respectively (Fig. 3). Then, if  $\text{MAX}(L_1, L_2, L_3, L_4) < G_0$  or  $\text{MAX}(R_1, R_2, R_3, R_4) < G_0$  (in case of JPEG) or  $\text{MAX}(L_1, L_2, L_3) < G_0$  or  $\text{MAX}(R_1, R_2, R_3) < G_0$  (in the case of H.264/AVC), the current row is marked as a boundary gap and is filtered by the 1D fuzzy filter (1)–(4). The length of the filter is five pixels, and the filter given in Eq. (2) is applied to all the pixels on the row or column within the block. The same process is also performed for horizontal artifacts. The spread parameter  $\sigma$  in Eq. (4) for 1D and 2D fuzzy filter is calculated based on the flowchart shown in Fig. 3.

4.1.3. Adaptive anisotropic diffusion

After applying deblocking to the image, the anisotropic diffusion equation, as described in Section 3, will be applied to the deblocked image. This section explains a discrete form of the anisotropic diffusion based on a numerical approach. Fig. 5 shows the effect of applying

Please cite this article as: E. Nadernejad, et al., Enhancing perceived quality of compressed images and video with anisotropic diffusion and fuzzy filtering, Signal Processing-Image Communication (2012), <http://dx.doi.org/10.1016/j.image.2012.12.001>

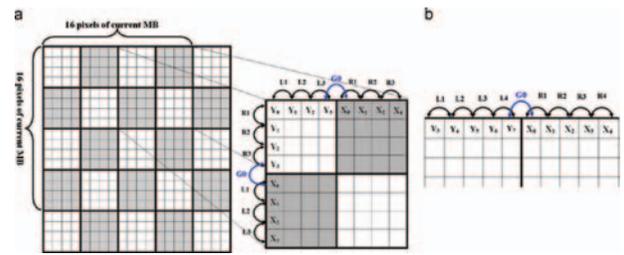
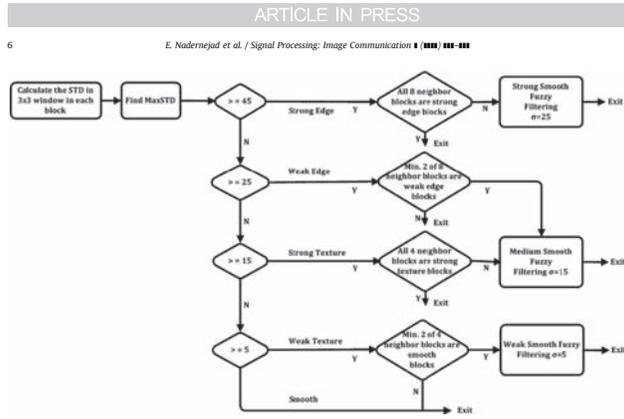


Fig. 4. Detecting the vertical boundary gap in a row across the vertical block boundary (a) for H.264/AVC video sequences and (b) for JPEG images and MJPEG video sequences.

the 1D and 2D anisotropic diffusion for removal of blocking artifacts in a JPEG compressed image. Comparison of these figures illustrate that 2D anisotropic diffusion can remove the blocking effect in the image, but in turn it causes blurring and smoothing effects in the output image. Fig. 6 depicts the pixel intensities along the 200th line of the image in Fig. 5. To avoid smoothing and blurring and increase the quality of images, only one-dimensional anisotropic diffusion is performed, rather than the two-dimensional diffusion, causing a blurring effect.

Nonlinear diffusion equations of the form given in Eq. (7) do not have an analytical or open form solution. These equations have to be solved using numerical methods. A finite difference based approach is one of the most common approaches used to solve this type of PDEs [30,35,37]. As mentioned in Section 3, the proposed anisotropic diffusion is an iterative adaptive algorithm. Based on a one-dimensional solution to Eq. (6)–(11), the discrete numerical solution of the proposed iterative adaptive diffusion algorithm is described below.

Please cite this article as: E. Nadernejad, et al., Enhancing perceived quality of compressed images and video with anisotropic diffusion and fuzzy filtering, Signal Processing-Image Communication (2012), <http://dx.doi.org/10.1016/j.image.2012.12.001>



Fig. 5. Comparison of 1D and 2D anisotropic diffusion equation applied to a JPEG compressed image for removing blocking artifact. All parameters are the same. (a) JPEG compressed image with blocking effect, (b) 1D anisotropic diffusion, (c) 2D anisotropic diffusion.

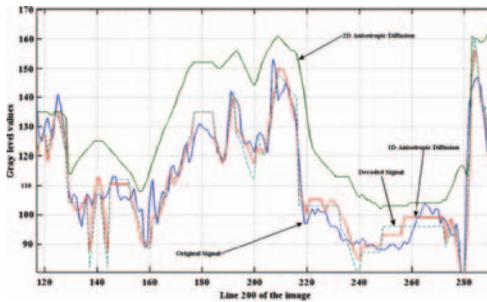


Fig. 6. The 200th line of the JPEG compressed image in Fig. 5, processed by 1D and 2D anisotropic diffusion.

1. Find the value of parameter  $k$  from Eqs. (9) and (10).
2. Let the time step be  $\Delta t$  and the spatial step be  $h$  in the  $x$  and  $y$  directions. Then, the time and space coordinates can be presented in discrete form as:

$$\begin{aligned} t &= n\Delta t, \quad n = 0, 1, 2, \dots; \\ x &= ih, \quad i = 1, 2, 3, \dots, M-1, \\ y &= jh, \quad j = 0, 1, 2, \dots, N-1, \end{aligned} \quad (13)$$

where  $h$  denotes pixel spacing in  $x$  and  $y$  directions. Pixels may be assumed to be located within the unit length (i.e.,  $h=1$ ) [30]. Let us approximate the image  $f(x, y, t)$  at time  $t$  by  $f_{ij}^n$ . Then, the left hand side of Eq. (6) can be written as:

$$\frac{\partial I}{\partial t} = \frac{f_{ij}^{n+1} - f_{ij}^n}{\Delta t}. \quad (14)$$

The size of the image is  $Mh \times Nh$  and the input image is  $f_{ij}^0 = I(ih, jh, 0)$ , derived from the deblocking phase based on 1D adaptive fuzzy filtering. Then, the final image can be

obtained by iterating the five-stage approach described below for the horizontal direction.

Stage I: The horizontal derivative approximations and the horizontal Laplacian approximations are computed for the frame:

$$\begin{aligned} \frac{f_{i+1,j}^n - f_{i,j}^n}{h}, \quad \frac{f_{i-1,j}^n - f_{i,j}^n}{h}, \\ \nabla^2 f_{ij}^n(hor) = \frac{(f_{i+1,j}^n + f_{i-1,j}^n - 2 \times f_{i,j}^n)}{h^2}, \end{aligned} \quad (15)$$

where  $h=1$ . The symmetric boundary conditions are used:

$$f_{-1,j}^n = f_{0,j}^n, \quad f_{M,j}^n = f_{M-1,j}^n, \quad j = 0, 1, 2, \dots, N-1. \quad (16)$$

Stage II: Computing the horizontal diffusion coefficient  $c(x,y;t)$  as in [30]:

$$c_{ij}^n = \exp\left(-\left[\frac{\nabla^2 f_{ij}^n(hor)}{k^2}\right]\right). \quad (17)$$

Please cite this article as: E. Nadernejad, et al., Enhancing perceived quality of compressed images and video with anisotropic diffusion and fuzzy filtering, Signal Processing-Image Communication (2012), <http://dx.doi.org/10.1016/j.image.2012.12.001>

## ARTICLE IN PRESS

8

E. Nadernejad et al. / Signal Processing: Image Communication 1 (2012) 125–131



Fig. 7. Proposed method for video frames.

Stage III: Computing the divergence of  $c(\cdot) \nabla I$ :

$$d_{ij}^c = \frac{1}{h} \left[ c_{i+1,j}^c (I_{i+1,j}^c - I_{i,j}^c) + c_{i,j}^c (I_{i,j}^c - I_{i,j+1}^c) \right], \quad (18)$$

with the symmetric boundary conditions:

$$d_{i,j}^c = d_{0,j}^c, \quad d_{i,j}^c = d_{i,N-1}^c, \quad j = 0, 1, 2, \dots, N-1. \quad (19)$$

Stage IV: The numerical approximation to the differential equation is given by:

$$I_{i,j}^{n+1} = I_{i,j}^n + \frac{\Delta t}{2} d_{i,j}^c. \quad (20)$$

Eq. (20) is equivalent to (6) in discrete form for the horizontal direction (in this work:  $\Delta t/2 = 0.125$ ).

Stage V: Check the stop criterion. For filtering, we use a fixed number of iterations, with 10 as the default value. When the stop criterion is reached, the algorithm terminates, otherwise it goes back to Stage I. After terminating the algorithm in the horizontal direction, the same algorithm is used in the vertical direction (Fig. 1). The optimal number of iterations in vertical and horizontal directions can be different if the artifacts are stronger in one direction than the other, since the best possible trade-off between smoothness of the edge versus blocking and ringing artifacts may be reached at different points.

#### 4.1.4. Two-dimensional fuzzy filtering

In Section 4.1.2, the standard deviation in a  $3 \times 3$  window was computed around each pixel and then the MaxSTD in each  $4 \times 4$  block or  $8 \times 8$  block was compared with a set of predetermined thresholds. In the following step, according to the type of the detected block in Eq. (12), and depending on its neighboring blocks, a 2D fuzzy filter Eq. (2) may be applied with adaptive spread parameter  $\sigma$ . For instance, if the detected block is labeled as a strong edge and not all of its surrounding blocks are strong edges, then the fuzzy filter will be applied with large spread parameter ( $\sigma = 25$ ); otherwise, no filtering is needed. In the case of strong or weak texture, the filtering decision is based only on four neighboring blocks (up, down, left and right). Fig. 3 illustrates the flowchart of fuzzy filtering parameter selection process for all block sizes in compressed images and videos. The method is inspired by [11], with a difference that we use a more sophisticated decision tree. The procedure for  $4 \times 4$  blocks and  $8 \times 8$  blocks is the same.

The fuzzy filter described here is used in the algorithm shown in Fig. 1 to reduce blocking (1D fuzzy) and ringing (2D fuzzy) artifacts, while retaining the sharpness of the edges. The main drawback of the isotropic fuzzy filter [13,41] for multi-dimensional signals, such as images, is that the signal is converted to a vector before filtering. The

direction between the pixels is ignored in this case. Blocking artifacts are either vertical or horizontal, whereas ringing artifacts occur along the edges of arbitrary direction. Therefore, it is expected that deringing performance would improve if the filter is applied adaptively according to the direction of the edges. The proposed algorithm is an adaptive algorithm, which accounts for both blocking and ringing artifacts, using both adaptive fuzzy filtering and anisotropic diffusion filter to further improve the quality.

#### 4.2. Adaptive fuzzy and anisotropic diffusion for artifact reduction in H.264/AVC video sequences

Fig. 7 illustrates a compound algorithm for artifact removal in video sequences, in particular video sequences compressed with H.264/AVC, represented in YUV 4:2:0 color space. Since the HVS processes brightness and chrominance information differently, several algorithms have been proposed for separate processing of luminance and chrominance channels [33,36,38].

In the proposed algorithm, we have examined different designs to find the best algorithm for deblocking and deringing in luminance and chrominance channels, and a hybrid algorithm where anisotropic diffusion is combined with fuzzy filtering is proposed, as shown in Fig. 7. In this algorithm, the chrominance components (U, V) of a video frame are first upsampled using bilinear interpolation to the same size with the luminance component (Y). To obtain higher quality, each Y frame component is enhanced by a directional spatial fuzzy filter. First, the initial amplitude for the spread parameter  $\sigma_0$  is derived, following the procedure explained in Section 4.1.4. Then, directional fuzzy filtering [14] is applied with spread parameter from Eq. (5). The spread parameter amplitude  $\sigma_A$  obtained from:

$$\sigma_A(i,j) = \sigma_0 \left( (1-\gamma) \times \frac{STD(i,j) - STD_{min}}{STD_{max} - STD_{min}} + \gamma \right), \quad (21)$$

where  $STD_{min}$  and  $STD_{max}$  are the minimum and maximum standard deviations found in the current frame, and  $\gamma$  is an offset parameter. In contrast, each U and V frame components are deblocked after upsampling using the spatial fuzzy filtering and anisotropic diffusion, as illustrated in Fig. 1 (Section 4.1).

#### 5. Performance evaluation methods

In order to assess the performance of the proposed method in comparison to other methods in the literature, we have used established full reference objective image and video quality metrics for test images and video

ARTICLE IN PRESS

E. Nadernejad et al. / Signal Processing-Image Communication 1 (2012) 121–131

9

sequences processed with different methods. In addition, we have performed a subjective assessment study to verify the results obtained from objective measurements. The metrics used are described shortly in this section.

### 5.1. Objective quality metrics

PSNR is a performance metric indicating the ratio between the maximum possible power of a signal and the power of the corrupting noise affecting the fidelity of the distorted signal. It is the most commonly used objective quality indicator for distorted digital images or video. When a color image or video sequence is processed, there are several options to compute overall PSNR. In the case of video representation in YUV color space, the luma (Y) component alone is often used for computing PSNR (referred as Y-PSNR), since luminance is more relevant for human perception than chrominance. It is also possible to compute PSNR over both luma and chroma components (YUV-PSNR), each component separately, or to use different weighting factors for luma and chroma components. Weight 0.8 is often used for luma and weight 0.1 for each of the chroma components [48]. In this paper, we study post-processing methods involving both luma and chroma components, and this is why we have chosen to use YUV-PSNR as one measure in our analysis (U- and V-components are upsampled to the same size as Y-component). When images in RGB color space are concerned, we have used average PSNR of R, G, and B components.

Structural similarity (SSIM) index can be used to measure the visual similarity of two images [49]. In SSIM, the total distortion is modeled as a combination of three different factors: loss of correlation, luminance distortion, and contrast distortion. Given that vector  $X = \{x_i | i = 1, 2, 3, \dots, N\}$  contains the intensities of all the  $N$  original pixels and  $Y = \{y_i | i = 1, 2, 3, \dots, N\}$  of the pixels in the degraded image, SSIM is defined as:

$$SSIM = \left( \frac{\sigma_{XY}}{\sigma_X \sigma_Y} \right)^\alpha \times \left( \frac{2\bar{X}\bar{Y}}{(\bar{X})^2 + (\bar{Y})^2} \right)^\beta \times \left( \frac{2\sigma_X \sigma_Y}{\sigma_X^2 + \sigma_Y^2} \right)^\gamma \tag{22}$$

$$= s(X, Y)^\alpha \times l(X, Y)^\beta \times c(X, Y)^\gamma$$

where  $\bar{X}$ ,  $\bar{Y}$  are the average pixel values of  $X$  and  $Y$ ,  $\sigma_X$  and  $\sigma_Y$  are standard deviations of  $X$  and  $Y$ ,  $\sigma_{XY}$  is the covariance between  $X$  and  $Y$ ,  $\alpha > 0$ ,  $\beta > 0$  and  $\gamma > 0$  are parameters defining the relative weights of the individual factors of quality. In order to simplify the expression, we have used values  $\alpha = \beta = \gamma = 1$  for our measurements. The dynamic range of SSIM is  $[-1, 1]$ . The highest possible value 1 is achieved if and only if  $y_i = x_i$  for all  $i = 1, 2, 3, \dots, N$ . In practical use, the local statistics for SSIM are often computed within a small sliding sampling window. As a result, a spatial SSIM index map is produced. In order to obtain a representative quality value for the whole image, Mean SSIM (MSSIM) value is then calculated by computing the mean value of the local SSIM values over the whole picture. Just as with PSNR, different approaches may be taken to compute MSSIM of a color signal. Most typically, MSSIM is computed for the luma (Y) component only, but chroma components may also be included. In this study, we have computed MSSIM for

Y-component only (using the Matlab script provided for the public by the authors of [49], using  $11 \times 11$  pixel blocks). For images in RGB color space, we have computed the average MSSIM of R, G, and B components.

To evaluate temporal artifacts like flickering, sum of squared differences (SSD) can be used to evaluate the temporal variation of the coding artifacts in video [1]. SSD is defined as:

$$SSD(t, i, j) = \frac{1}{N} \sum_{m, n \in \{i, j\}} (D(t, m, n) - D(t-1, m, n))^2 \tag{23}$$

where  $N$  is the number of frames,  $t$  is the frame index, and  $D$  is the difference between the original frame  $X$  and the compressed frame  $Y$ :

$$D(t, m, n) = X(t, m, n) - Y(t, m, n) \tag{24}$$

Larger SSD values correlates with higher flickering intensity.

### 5.2. Subjective quality assessment

Even though objective metrics for visual quality have shown a relatively good match with subjective perception of quality in many studies, subjective quality assessment is still considered the ultimate approach to evaluate visual quality. Several different subjective quality assessment methods have been proposed in the literature and many of them have even been standardized [50]. The methods can be classified into many ways, for example according to the number of stimuli (single, double, multiple), task (direct or comparative rating, rank ordering) or scaling (discrete, continuous, binary).

Most typically, subjective quality assessment involves quality rating, and the final result is expressed in terms of mean opinion score (MOS), which is the average of the scores by individual test subjects. Rating is often considered problematic, since different individuals may interpret vocabulary and intervals of the rating scale differently, and ratings may be influenced by the attractiveness of the content [51]. This is why formation of MOS out of individual scores may require some processing, such as removal of outliers and systematic personal bias. In order to make results more reliable, many subjective evaluation procedures involve comparative rating, where the test stimulus is compared against the (typically distortion free) reference stimulus.

When post-processing algorithms are compared, the visual differences are often small, and subjects can only identify those differences clearly when different stimuli are shown concurrently. This is why direct MOS rating does not suit our study well. Another option could be to use paired comparisons [52], but if all the possible pairs are tested, it would lead to a large number of test cases. This is why we have adopted a rank ordering method, where all the stimuli are shown at the same time and test subjects are asked to rank them in the order of preference. To our knowledge, this method is not commonly used for subjective video quality assessment, but a similar method has been described for comparing still images in [53]. The video sequences are looping, and therefore the test subjects have the opportunity to observe the subtle

Please cite this article as: E. Nadernejad, et al., Enhancing perceived quality of compressed images and video with anisotropic diffusion and fuzzy filtering, Signal Processing-Image Communication (2012), <http://dx.doi.org/10.1016/j.image.2012.12.001>

## ARTICLE IN PRESS

10

E. Nadernejad et al. / Signal Processing: Image Communication 1 (2012) 100–108

differences carefully, in spite of short duration of the sequences.

In our practical subjective experiments, we have used a 55 in. high quality liquid crystal display with LED backlight delivered by Bang&Olufsen, aimed to be used as a television screen. Since the screen is capable of full high definition resolution, it is possible to accommodate several CIF resolution video sequences on the screen at the same time and play them synchronized. The large size of the screen allows viewing the details of sequences clearly. The viewing distance was 2.0 m, and the physical height of a single CIF resolution video on screen is 18 cm.

## 6. Experimental results

Tests have been performed to demonstrate the effectiveness of the two proposed algorithms on JPEG images (Fig. 1) and MJPEG and H.264/AVC video sequences (Fig. 7), respectively. The qualities of the different approaches have been compared in terms of PSNR, MSSIM and subjective visual quality. For comparison, the artifact removing methods proposed by Zhai et al. [2], Kim [3], Chen et al. [4], Liu et al. [5], Vo et al. (fuzzy-based methods) [11,13,14], Tai et al. [16], Yao (PDE-based algorithm) [9], Chebbo et al. [43], and, have been tested. The 5/3 MCTF method, which is proposed by Chen et al. [17], has been also simulated. We have tested the implementations for Chen, Liu and 5/3 MCTF methods publicly available in the Internet [54]. Kim's method, Tai's method, Zhai's method, Yao's method, Chebbo's method, isotropic fuzzy method, 2D fuzzy method and directional fuzzy method have been implemented.

### 6.1. Enhancement of compressed images

To evaluate the efficacy of the proposed post-processing technique for reducing blocking and ringing artifacts in still images (Fig. 1), we have conducted experiments with test images from video sequences, compressed using the JPEG image compression algorithm. The performance of the proposed method has been compared against Chen's method, Liu's method, Kim's method, Chebbo's method, Tai's method, adaptive 2D fuzzy filter [13], Zhai's method, directional fuzzy filter [14] and isotropic 2D fuzzy filter with constant parameters [11]. For the directional fuzzy method, only the non-edge pixels (with  $G > 180$  in [14]) are filtered in order

to avoid smoothing the real edges of the image and all parameters were chosen from Ref. [14]. All the parameters in Section 5 are chosen experimentally using a wide selection of images to achieve the optimal visual quality. In the proposed algorithm, there are some parameters, which need to be initialized. The parameter  $k$  in Eq. (7) is chosen by using Eq. (9). Because the proposed algorithm is iterative, we have used a wide range of different images with highly varying texture, luminance, etc., to train the algorithm and find the optimal number of iterations. In the training phase, PSNR of the compressed and post-processed image is measured after each iteration round. It is assumed that PSNR increases until the optimal number of iterations is reached, and then starts decreasing. Therefore, the stop criterion during training is:

$$\text{PSNR}(n+1) < \text{PSNR}(n), \quad (25)$$

where  $n$  is the number of iterations. In average, the proposed method achieved the best PSNR with 10 iterations for both horizontal and vertical directions. This default value is used unless otherwise stated.

Parameter  $\gamma$  defined in Eq. (21) controls the balance between artifact removal in the flat regions and keeping the details in the high activity regions. Parameters  $\alpha$  and  $\beta$  in (5) are used to adjust the relative filtering strength between the gradient and tangent directions of the edges. We have used the values for parameters in Eqs. (5) and (22) from [14], giving, i.e.  $\gamma=0.5$ ,  $\alpha=0.5$  and  $\beta=3.5$ , and  $\sigma_0$  computed as shown in the flowchart in Fig. 3. The set  $\omega$  of neighboring pixels in Eq. (1) and the spatial window  $W$  size are set to  $5 \times 5$ . We have compressed several CIF resolution video sequences using Motion JPEG with scaling factor 4 for the quantization. The test set includes different types of images (frames), converted to RGB format, from the following video contents: Silent, Foreman, Mobile, Paris, News, and Mother. The average PSNR and MSSIM values for the six different sequences with 50 frames (images) are listed in Tables 1 and 2, respectively. PSNR and MSSIM values for each frame are the equally weighted averages computed from the R, G, and B components. To validate the use of predefined number of iterations obtained by training with a different set of images, we have also included the results with the optimal numbers of iterations (25) for different images.

To demonstrate the visual quality, the results obtained with different artifact removal techniques on a compressed 6th frame in Mother and Daughter are shown in Fig. 8. In this experiment, the spread parameter of the proposed method

**Table 1**  
Comparison of different methods by average RGB-PSNR (dB) (first column, JPEG and other methods, 2=Chen's method, 3=Liu's Method, 4=Kim's Method, 5=Chebbo's Method, 6=Zhai's method, 7=2 D Fuzzy, 8=Tai's method, 9=directional [9], 10=Yao's method [49], 11=Isotropic, 12=proposed with iteration=10, 13=proposed with optimum number of iterations (the number of iterations is shown inside the parentheses).

PSNR	1	2	3	4	5	6	7	8	9	10	11	12	13
News	25.56	26.03	25.86	26.08	26.11	26.13	25.97	26.08	26.17	26.07	25.56	26.16	<b>26.19</b> (11)
Silent	25.53	26.12	25.96	26.11	26.22	26.32	26.19	26.27	26.22	26.29	25.54	26.38	<b>26.43</b> (11)
Foreman	25.33	25.86	25.91	25.81	26.05	26.21	25.98	26.11	26.05	26.15	25.43	26.22	<b>26.26</b> (9)
Mobile	21.25	21.17	21.33	21.48	21.51	21.73	21.76	21.65	21.75	21.62	21.28	<b>21.96</b>	21.96 (10)
Mother	27.81	28.35	28.19	28.46	28.56	28.51	28.66	28.75	28.41	28.53	28.81	28.87	<b>28.90</b> (11)
Paris	21.89	22.35	22.25	22.37	22.42	22.51	22.50	22.53	22.43	22.45	22.41	22.68	<b>22.71</b> (9)
Ave Gain	0.419	0.358	0.489	0.584	0.673	0.614	0.669	0.61	0.58	0.227	0.818	<b>0.847</b>	

Please cite this article as: E. Nadernejad, et al., Enhancing perceived quality of compressed images and video with anisotropic diffusion and fuzzy filtering, Signal Processing-Image Communication (2012), <http://dx.doi.org/10.1016/j.image.2012.12.001>

**Table 2**  
Comparison of different methods by average RGB-MSSIM (first column, JPEG and other methods, 2=Chen's method, 3=Liu's method, 4=Kim's method, 5=Chebbo's method, 6=Zhai's method, 7=2D Fuzzy, 8= Tai's method, 9=directional [9], 10=Yao's method [49], 11=Isotropic, 12=proposed method.

	1	2	3	4	5	6	7	8	9	10	11	12
News	0.489	0.508	0.503	0.505	0.510	0.514	0.515	0.517	0.513	0.513	0.509	<b>0.520</b>
Silent	0.413	0.422	0.441	0.438	0.445	0.446	0.444	0.448	0.444	0.443	0.434	<b>0.452</b>
Foreman	0.470	0.486	0.495	0.492	0.528	<b>0.531</b>	0.503	0.510	0.499	0.521	0.494	0.518
Mobile	0.462	0.458	0.469	0.471	0.475	0.483	0.482	0.489	0.480	0.478	0.475	<b>0.491</b>
Mother	0.495	0.510	0.505	0.515	0.512	0.508	0.523	0.531	0.522	0.510	0.517	<b>0.533</b>
Paris	0.444	0.462	0.460	0.461	0.463	0.469	0.470	0.473	0.466	0.466	0.461	<b>0.479</b>
Ave Gain	0.014	0.015	0.018	0.023	0.029	0.027	0.032	0.025	0.026	0.020	0.038	<b>0.038</b>

has been calculated using the method in Fig. 3. Compared to the compressed full frame in Fig. 8a (PSNR=31.02, MSSIM=0.839) and the enhanced image using Chen's method (Fig. 8b, PSNR=31.83, MSSIM=0.869), Liu's method (Fig. 8c, PSNR=31.61, MSSIM=0.862), Kim's method (Fig. 8d, PSNR=31.69, MSSIM=0.861), Chebbo's method (Fig. 8e, PSNR=31.65, MSSIM=0.864), the 2D Fuzzy filter method (Fig. 8f, PSNR=31.77, MSSIM=0.865), Zhai's method (Fig. 8g, PSNR=31.64, MSSIM=0.862), Tai's method (Fig. 8h, PSNR=31.84, MSSIM=0.869), Yao's method (PSNR=31.78, MSSIM=0.865), the enhanced image using the proposed method in Fig. 8i (PSNR=32.10, MSSIM=0.872) achieved a clear improvement in terms of both PSNR and MSSIM.

Comparing to the compressed image in Fig. 8a, all of the post-processing methods can remove most of the blocking and ringing artifacts. However, these methods introduce other types of artifacts, such as blurring at the dominant edges, and loss of details. For example, the DCT-based low-pass filtering technique proposed by Chen (Fig. 8b) is able to suppress some of the ringing artifacts, but in turn it causes a substantial blurriness in the processed image. Liu's method (Fig. 8c) and Kim's method (Fig. 8d) are able to retain some of the sharpness, but is not able to reduce the ringing artifacts. Chebbo's method (Fig. 8e) shows ringing around the edges, as well as blocking, especially within the facial area. 2D fuzzy filter (Fig. 8f) also shows some visible ringing and blocking effects. The Zhai's method (Fig. 8g) and Tai's method (Fig. 8h) cannot remove blocking artifacts successfully, and they introduce some smoothing effect. The frame enhanced with the proposed algorithm shows the best quality, compared to the other algorithms. It efficiently removes the blocking and ringing artifacts, but still keeps the details and the sharpness of the edges.

6.2. Enhancement of compressed video sequences

To demonstrate the advantage of the proposed method designed for video sequences (Fig. 7), the experiments in this section are performed on MJPEG and H.264/AVC coded sequences, represented in YUV format.

6.2.1. Enhancement for MJPEG video sequences

In MJPEG, each frame of a video sequence is compressed separately using the JPEG standard. In the practical implementation, the parameter  $K$  in Eq. (4) is chosen as  $K=1$  [13]. The sizes for  $\sigma$  and the spatial window are  $5 \times 5$  pixels.

Fig. 9 compares the enhanced frame obtained using different post-processing methods, including the proposed algorithm, applied on the 35th frame of the Mobile video sequence. The enhanced frame obtained by the proposed method for video artifact reduction (Fig. 9f) shows significantly reduced ringing artifacts and better color quality than the other algorithms, including the 2D fuzzy method [13]. The visual improvement obtained with the proposed scheme is much more noticeable when the processed frames are played in a sequence, since the proposed method produces a video of smoother quality with significantly reduced artifacts.

Figs. 10 and 11 compare the PSNR, and MSSIM values of all the tested methods for 45 frames of the Mobile sequence. In this section, we use YUV-PSNR and Y-MSSIM, unless stated otherwise. The plots clearly demonstrate that the proposed method for video sequences achieves a consistent average PSNR gain of about 0.55 dB relative to the compressed frames without post-processing, and about 0.23 dB relative to the frames enhanced with the 2D fuzzy filter method. The respective MSSIM gains are about 0.02 and 0.05.

6.2.2. Enhancement for H.264/AVC video sequences

In order to demonstrate that the proposed method is beneficial also for H.264/AVC video compression, which tend to have less artifacts compared to previous method, further experiments were performed with sequences compressed using H.264/AVC video compression standard. For this purpose, we have compressed several CIF resolution video sequences with H.264/AVC reference encoder using GOP length 12 (prediction structure IBBPBB...) and fixed QP of 45. The standard in-loop deblocking filter was enabled. In the post-processing phase, the offset parameter  $\gamma$  in Eq. (21) for the directional fuzzy filtering (Fig. 7) was set to 0.5. This value ( $\gamma$ ) has been chosen experimentally to get the best visual quality for a wide range of sequences. The PSNR values for the compressed sequences with and without post-processing are given in Table 3. As the results show, PSNR is improved for luma and both chroma components in all cases when the proposed method is used. However, the gain is less significant for sequences with complex textures, such as Coastguard. The average gains are 0.12 dB, 0.21 dB, and 0.26 dB for the Y, U and V-components, respectively.

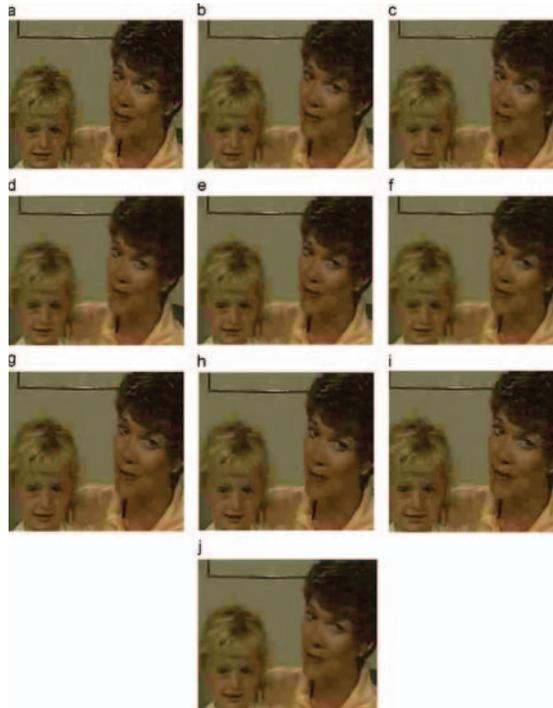
To study the performance more comprehensively, we have analyzed the Foreman sequence compressed using the prediction structure of IBBPBBBPB at 132 kbps (rate

Please cite this article as: E. Nadernejad, et al., Enhancing perceived quality of compressed images and video with anisotropic diffusion and fuzzy filtering, Signal Processing-Image Communication (2012), <http://dx.doi.org/10.1016/j.image.2012.12.001>

## ARTICLE IN PRESS

12

E. Nadernejad et al. / Signal Processing: Image Communication 1 (2012) 132–144



**Fig. 8.** The comparison of filtered results on zoomed images: (a) the compressed image-4Q (PSNR=31.02, MSSIM=0.839), (b) Chen's method (PSNR=31.83, MSSIM=0.869); (c) Liu's method (PSNR=31.61, MSSIM=0.862), (d) Kim's method (PSNR=31.69, MSSIM=0.861), (e) Chebbo's method (PSNR=31.65, MSSIM=0.864), (f) 2D fuzzy filter method (PSNR=31.77, MSSIM=0.865), (g) Zhai's method (PSNR=31.64, MSSIM=0.862), (h) Tai's method (PSNR=31.84, MSSIM=0.869), (i) Yao's method (PSNR=31.78, MSSIM=0.865) and (j) the proposed method (PSNR=32.10, MSSIM=0.872).

control is used instead of fixed QP) in more detail. Fig. 12 shows the third frame from the compressed Foreman sequence (Fig. 11b), the Kim's method (Fig. 12c), the Chebbo's method (Fig. 12d), the Haar MCTF method (Fig. 12e), the Zhai's method (Fig. 12f), the 2D fuzzy

filtering method (Fig. 12g), the Tai's method (Fig. 12h), the Yao's method (Fig. 12i) and enhanced frame with the proposed method (Fig. 12j).

In this experiment, the proposed algorithm improves PSNR and MSSIM, compared to the PSNR and MSSIM with

Please cite this article as: E. Nadernejad, et al., Enhancing perceived quality of compressed images and video with anisotropic diffusion and fuzzy filtering, Signal Processing-Image Communication (2012), <http://dx.doi.org/10.1016/j.image.2012.12.001>



**Fig. 9.** Comparison of filter results for MPEG sequences (35th frame). (a) Compressed (PSNR=23.03, MSSIM = 0.786), (b) Chen's method (PSNR=22.58, MSSIM=0.774); (c) Liu's method (PSNR=22.88, MSSIM=0.782); (d) Kim's method (PSNR=23.28, MSSIM=0.794); (e) Chebbo's method (PSNR=23.13, MSSIM=0.795); (f) 2D fuzzy filter method (PSNR=23.28, MSSIM=0.804); (g) Zhai's method (PSNR=23.23, MSSIM=0.794); (h) Tai's method (PSNR=23.42, MSSIM=0.798); (i) Yao's method (PSNR=23.49, MSSIM=0.784) and (j) The proposed method (PSNR=23.56, MSSIM=0.819).

Chen's method, Liu's method, Kim's method, Chebbo's method, Haar MCTF method, Zhai's method, 2D fuzzy method, Tai's method, and in-loop filtering (H.264/AVC video sequences). This improvement is consistent for the Foreman sequence, as verified by the PSNR and MSSIM curves shown in Figs. 13 and 14. The average results are shown in Table 4. The average PSNR and MSSIM performance of the proposed method is better than other

methods. Visual inspection shows that compared to H.264/AVC frames, all the post-processing algorithms significantly reduce the ringing and blocking artifacts. However, the proposed method maintains the sharpness of edges better than the 2D fuzzy method [13]. This improvement indicates a more pleasant visual appearance.

To evaluate the effectiveness of the proposed algorithm in the temporal dimension, SSD (Eq. (23)) is also computed

Please cite this article as: E. Nadernejad, et al., Enhancing perceived quality of compressed images and video with anisotropic diffusion and fuzzy filtering, Signal Processing-Image Communication (2012), <http://dx.doi.org/10.1016/j.image.2012.12.001>

ARTICLE IN PRESS

14

E. Nadernejad et al. / Signal Processing: Image Communication 1 (2012) 133–144

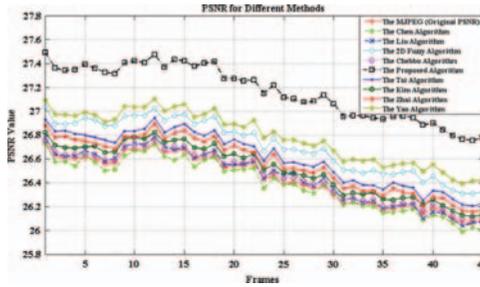


Fig. 10. Comparison on PSNR of simulated methods for mobile sequence (average PSNR: MJPEG=26.43, Chen's method=26.30, Liu's method=26.45, Kim's method=26.51, Chebba's method=26.46, Zhai's method=26.56, 2D fuzzy=26.71, Tai's Method=26.61, Yao's Method=26.78, Proposed=27.17).

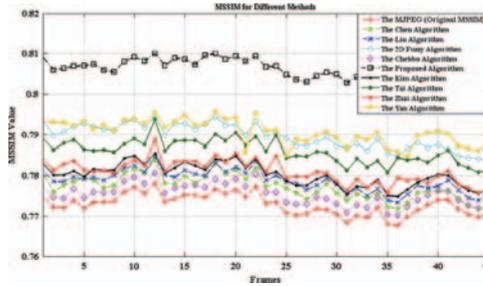


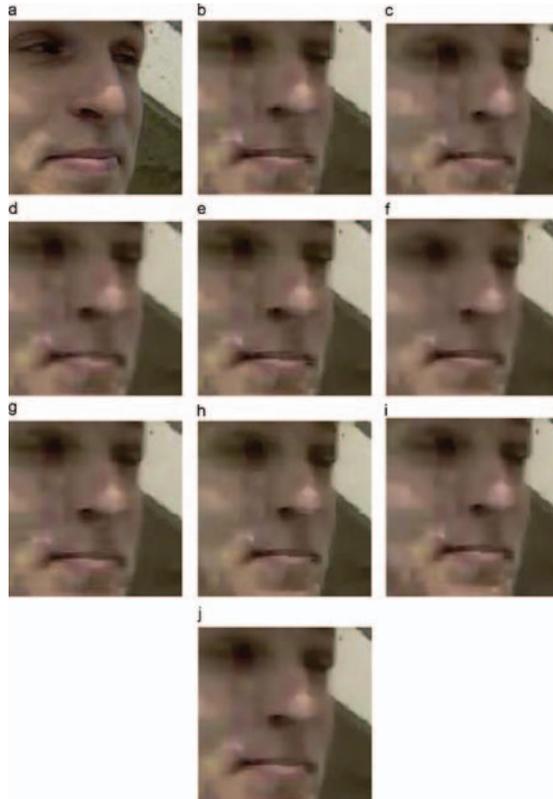
Fig. 11. Comparison on MSSIM of simulated methods for Mobile sequence (average MSSIM: MJPEG=0.773, Chen's method=0.777, Liu's method=0.779, Kim's Method=0.780, Chebba's method=0.775, Tai's Method=0.786, Zhai's method=0.781, 2D fuzzy=0.789, Yao's Method=0.791, Proposed=0.806).

Table 3

Y, U, and V-PSNR for H.264/AVC compressed sequences of 100 frames with and without the proposed post-processing.

Sequence (CIF, QP=45)	H.264/AVC			Proposed			Gain (ΔPSNR)		
	Y	U	V	Y	U	V	Y	U	V
Coastguard	25.32	38.89	39.35	25.34	38.98	39.68	0.02	0.08	0.31
Mobile-calendar	22.81	31.45	31.87	22.90	31.46	31.88	0.09	0.01	0.01
Akiyo	30.34	36.07	38.77	30.48	36.37	39.23	0.14	0.30	0.46
Hall monitor	27.94	36.02	38.01	28.15	36.27	38.20	0.21	0.26	0.19
Mother and daughter	30.38	38.48	39.09	30.46	39.05	39.62	0.08	0.57	0.53
Silent	27.54	35.33	36.77	27.64	35.41	37.02	0.10	0.08	0.25
News	27.62	35.32	36.63	27.77	35.53	36.81	0.15	0.21	0.18
Foreman	28.00	36.98	37.84	28.17	37.14	38.00	0.17	0.15	0.16
<b>Average gain</b>							<b>0.12</b>	<b>0.21</b>	<b>0.26</b>

Please cite this article as: E. Nadernejad, et al., Enhancing perceived quality of compressed images and video with anisotropic diffusion and fuzzy filtering, Signal Processing-Image Communication (2012), <http://dx.doi.org/10.1016/j.image.2012.12.001>



**Fig. 12.** Comparison of filter results for H.264/AVC sequences (frame no. 3 zoomed): (a) original frame, (b) compressed with in-loop filtering (PSNR = 32.30, MSSIM = 0.869), (c) Kim's method (PSNR = 32.32, MSSIM = 0.872), (d) Chebbo's method (PSNR = 32.32, MSSIM = 0.870), (e) 5/3 Haar MCTF method (PSNR = 32.11, MSSIM = 0.873), (f) Zhai's method (PSNR = 32.45, MSSIM = 0.873), (g) 2D fuzzy filter method (PSNR = 32.13, MSSIM = 0.872), (h) Tai's method (PSNR = 32.44, MSSIM = 0.874), (i) the Yao's method (PSNR = 32.42, MSSIM = 0.873), (j) the proposed method (PSNR = 32.57, MSSIM = 0.875).

Please cite this article as: E. Nadernejad, et al., Enhancing perceived quality of compressed images and video with anisotropic diffusion and fuzzy filtering, Signal Processing-Image Communication (2012), <http://dx.doi.org/10.1016/j.image.2012.12.001>

## ARTICLE IN PRESS

16

E. Nadernejad et al. / Signal Processing: Image Communication 1 (2012) 133–144

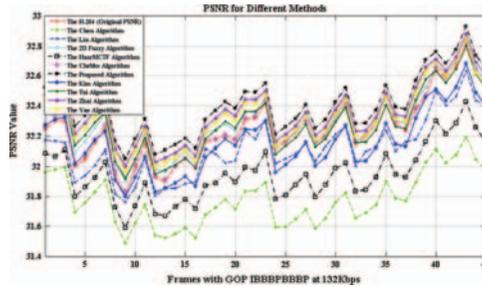


Fig. 13. Comparison on PSNR of simulated methods for Foreman sequence.

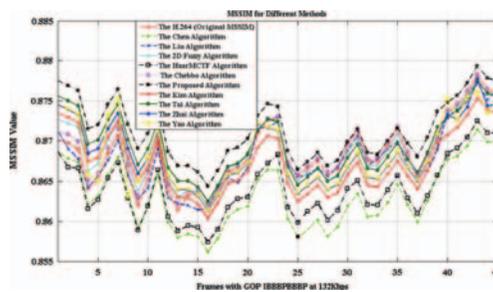


Fig. 14. Comparison on MSSIM of simulated methods for Foreman sequence.

for the processed Foreman sequence. The average results are included in Table 4. The results show that Chen's, Liu's and Haar's algorithms increase SSD thus indicating more flickering compared to the original compressed sequence, whereas the other algorithms reduce the SSD and therefore likely also reduce the flickering. The proposed algorithm shows the lowest level of flickering artifacts as measured by SSD.

The video sequence compressed with H.264/AVC (in-loop deblocking filter enabled) has fewer artifacts when enhanced using the proposed method, compared to the compressed sequence enhanced with the fuzzy method [11,14]. The PSNR improvement obtained with the proposed algorithm applied to the Foreman sequence compressed with different bitrates are shown in Fig. 15. The proposed algorithm yields more than 0.33 dB

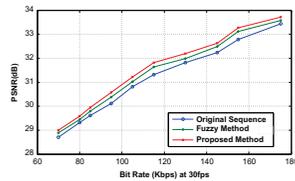
improvement, compared to the fuzzy method giving 0.21 dB improvement for bitrates from 70 kbps to 170 kbps. Also, visual inspection indicates higher quality when the proposed method is used. The proposed algorithm is an iterative algorithm, and compared to the other spatial algorithms it is therefore more complex. The proposed method does not require a motion compensation stage and spatiotemporal filtering, like 5/3 MCTF (Haar's method) described in [17]. Nevertheless, the most relevant comparison point to our method is the 2D fuzzy method proposed by Vo and Vetro [13], and our method shows steadily better performance, measured by both PSNR and MSSIM. Extending the algorithm by using temporal filtering as in [13] should also improve the performance of the proposed method.

Please cite this article as: E. Nadernejad, et al., Enhancing perceived quality of compressed images and video with anisotropic diffusion and fuzzy filtering, Signal Processing-Image Communication (2012), <http://dx.doi.org/10.1016/j.image.2012.12.001>

**Table 4**

The average PSNR, MSSIM and flickering metric (SSD) of compressed Foreman sequence: 1= H.264/AVC without post-processing, 2=Chen's method, 3= Liu's Method, 4 =Kim's method, 5= Chebbo's method, 6= Haar MCTF method, 7=Guangtao's method, 8= 2D fuzzy, 9= Tai's method, 10= Yao's method, 11=proposed method, respectively.

	1	2	3	4	5	6	7	8	9	10	11
PSNR	32.24	31.78	32.11	32.27	32.26	31.95	32.35	32.28	32.31	32.32	<b>32.46</b>
MSSIM	0.868	0.863	0.868	0.869	0.869	0.869	0.868	0.869	0.869	0.868	<b>0.871</b>
SSD	59.44	64.25	60.27	59.43	59.18	62.37	59.10	57.55	58.66	56.31	<b>53.81</b>



**Fig. 15.** Comparison of PSNR with different bit-rates of the Foreman sequence.

**6.2.3. Subjective evaluation of MJPEG and H.264/AVC video sequences**

In order to verify the objective measurements, a subjective comparison study based on rank ordering of simultaneously played video sequences as described in Section 5.3 was performed by using 16 test subjects. In this phase, we have included the "Mobile Calendar" sequence compressed with MJPEG and "Foreman" sequence compressed with H.264/AVC. In order to facilitate test arrangements and limit the number of tested sequences in each comparison, the subjective tests have been performed in two phases. In the first phase, we compared the baseline (MJPEG or H.264/AVC) against the proposed method and four other methods: Liu, Chen, Chebbo and 2D fuzzy. For H.264/AVC test also Haar method was included. The results of the first phase showed best performance for the proposed method, 2D Fuzzy and Chebbo methods. The other methods showed substantially worse performance. For the second phase, we have chosen the two best performing methods (2D fuzzy and the proposed method) from the first phase to be compared against three other methods not used in the first test: Kim, Tai and Zhai. The results from the second phase are summarized in Tables 5 and 6. As the results show, the average ranking shows the best performance for the proposed method in both cases. The closest competitor is 2D fuzzy method, like in the objective results.

With some exceptions for methods giving close results, the subjective ranking is reasonably well in line with rankings based on PSNR and MSSIM, suggesting that both metrics can be used for a rough evaluation of the relative performance of different algorithms for artifact removal. The visible differences between most of the sequences are rather small, and therefore some uncertainty in the

**Table 5**

Subjective ranking of different methods (Mobile calendar compressed with MJPEG), compared with PSNR and MSSIM rankings and gains ( $\Delta$ ) relative to MJPEG coding.

Method	Mean subjective rank	Rank PSNR ( $\Delta$ PSNR)	Rank MSSIM ( $\Delta$ MSSIM)
MJPEG	5.563	6. (23.03 dB)	6. (0.786)
Kim	4.188	5. (+0.98 dB)	3. (+0.005)
Tai	4.125	3. (+0.17 dB)	4. (+0.013)
Zhai	3.375	4. (+0.13 dB)	5. (+0.003)
2D fuzzy	2.250	2. (+0.24 dB)	2. (+0.018)
Proposed	1.000	1. (+0.53 dB)	1. (+0.033)

**Table 6**

Subjective ranking of different methods (Foreman compressed with H.264/AVC), compared with PSNR and MSSIM rankings and gains ( $\Delta$ ) relative to H.264/AVC with in-loop filtering.

Method	Mean subjective rank	Rank PSNR ( $\Delta$ PSNR)	Rank MSSIM ( $\Delta$ MSSIM)
H.264/AVC	4.750	6. (32.24 dB)	6. (0.8684)
Kim	4.688	5. (+0.03 dB)	3. (+0.0022)
Tai	4.188	3. (+0.07 dB)	2. (+0.0017)
2D fuzzy	3.813	4. (+0.05 dB)	4. (+0.0004)
Zhai	1.875	2. (+0.11 dB)	5. (+0.0000)
Proposed	1.250	1. (+0.17 dB)	1. (+0.0029)

results is expected. It should be noted that PSNR and MSSIM estimate the quality on per-frame basis, omitting temporal artifacts, such as flickering. The results indicate that flickering does not have a dominating role in our experiments, but in sequences with significant temporal artifacts, quality indicators such as PSNR and MSSIM would most likely give less reliable results.

For MJPEG, all post-processing methods are visually ranked better than direct decoding without post-processing. For H.264/AVC, 2D fuzzy, Chebbo and the proposed methods are ranked better, both visually, and by PSNR and MSSIM. 2D fuzzy and Chebbo have a similar average ranking, while the proposed method is preferred.

**7. Conclusions**

In this paper, we have presented two effective algorithms for image and video artifact removal, based on combining adaptive fuzzy filtering and directional anisotropic diffusion. The novel method improves the performance by adapting to the activity and direction between pixels at edges, and therefore preserves the visually

Please cite this article as: E. Nadernejad, et al., Enhancing perceived quality of compressed images and video with anisotropic diffusion and fuzzy filtering, Signal Processing-Image Communication (2012), <http://dx.doi.org/10.1016/j.image.2012.12.001>

## ARTICLE IN PRESS

18

E. Nadernejad et al. / Signal Processing: Image Communication 1 (2012) 138–144

essential elements, such as edges and fine textures, better than the traditional methods used for removing ringing and blocking artifacts. We have shown that the proposed algorithms improves the visual quality of compressed images and videos in terms of PSNR, SSD and, MSSIM, when compared against other well-known spatial post-processing methods in the literature. The objective results have been confirmed by subjective quality assessment based on rank ordering. The proposed adaptive scheme can be applied to images and video sequences compressed with several different standards, such as JPEG, MJPEG and H.264/AVC, and it shows robust performance on different types of contents compressed with these standards, even for H.264/AVC using in-loop filtering.

## References

- [1] X. Fan, W. Gao, Y. Lu, D. Zhao, Flicker reduction in all intra frame coding. Joint Video Team of ISO/IEC MPEG and ITU-T VCEG, JVT-E070, Oct. 2002.
- [2] G. Zhai, W. Zhang, X. Yang, Y. Xu, Efficient deblocking with coefficient regularization, shape adaptive filtering and quantization constraint, *IEEE Transactions on Multimedia* 10 (5) (2008) 735–745, Aug.
- [3] C. Kim, Adaptive post-filtering for reducing blocking and ringing artifacts in low bit-rate video coding, *Signal Processing: Image Communication* 17 (2002) 525–535.
- [4] T. Chen, H. Wu, B. Qiu, Adaptive postfiltering of transform coefficients for the reduction of blocking artifacts, *IEEE Transactions on Circuits and Systems for Video Technology* 11 (2001) 504–602.
- [5] S. Liu, A. Bovik, Efficient DCT-domain blind measurement and reduction of blocking artifacts, *IEEE Transactions on Circuits and Systems for Video Technology* 12 (2002) 1139–1149.
- [6] Y. Yang, N.P. Galatsanos, Removal of compression artifacts using projections onto convex sets and line process modeling, *IEEE Transactions on Image Processing* 6 (1997) 1345–1357.
- [7] Z.L. Xu, S.L. Xie, A deblocking algorithm using anisotropic diffusion based on HVS, *Computer Engineering* 32 (4) (2006) 10–12.
- [8] E. Choi, M.G. Kang, Deblocking algorithm for DCT-based compressed images using anisotropic diffusion, in: *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Proc. (ICASSP'03)*, vol. III, 2003, pp. 717–720.
- [9] S. Yao, K.P. Lim, X. Lin, S. Rahardja, A post-processing algorithm using histogram-driven anisotropic diffusion, in: *Proc. IEEE International Symposium on Circuits and Systems (ISCAS'05)*, vol. 5, pp. 4233–4236, 2005.
- [10] S. Oguz, Y. Hu, T. Nguyen, Image odng ringing artifact reduction using morphological post-filtering, in: *Proc. IEEE Int. Work. Multimedia Signal Proc. (MMS'08)*, pp. 628–633, 1998.
- [11] Y. Nie, H.-S. Kong, A. Vetro, H. Sun, K.E. Barner, Fast adaptive fuzzy post-filtering for coding artifacts removal in interlaced video, in: *Proc. of IEEE Int. Conf. Acoustics, Speech, and Signal Proc. (ICASSP'05)*, 2005, pp. 993–996.
- [12] A. Leonardis, Y. Tomomura, T. Nakachi, and P. Cosman, Flicker suppression in JPEG2000 using segmentation-based adjustment of block truncation lengths, in: *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Proc. (ICASSP'07)*, vol. 1, 2007, pp. 1117–1120.
- [13] D. Vo, T. Nguyen, S. Yea, A. Vetro, Adaptive fuzzy filtering for artifact reduction in compressed images and videos, *IEEE Transactions on Image Processing* 18 (6) (2009) 1166–1178, May.
- [14] D. Vo, T. Nguyen, S. Yea, and A. Vetro, Edge-based directional fuzzy filter for compression artifact reduction in JPEG images, in: *Proc. IEEE Int. Conf. Image Proc. (ICIP'08)*, 2008, pp. 797–800.
- [15] N. Yao, E.B. Kermeth, Fuzzy weighted median filters, in: *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Proc. (ICASSP'02)*, vol. 2, 2002, pp. 1125–1128.
- [16] S.-C. Tai, Y.-Y. Chen, S.-F. Sheu, Deblocking filter for low bit rate MPEG-4 video, *IEEE Transactions on Circuits and Systems for Video Technology* 15 (6) (2005) 733–741, Jun.
- [17] C.Y. Chen, C.T. Huang, Y.H. Chen, S.Y. Chien, L.G. Chen, System analysis of VLSI architecture for 5/3 and 1/3 motion-compensated temporal filtering, *IEEE Transactions on Image Processing* 54 (10) (2006) 4004–4014.
- [18] G. Zhai, W. Lin, J. Cai, X. Yang, W. Zhang, Efficient quadtree based block-shift filtering for deblocking and deringing, *Journal of Visual Communication and Image Representation* 20 (8) (2009) 595–607.
- [19] S.-H. Shin, K.-H. Park, Y.-J. Chai, T.-Y. Kim, Variable block-based deblocking filter for H.264/AVC on low-end and low-bit rates terminals, *Signal Processing: Image Communication* 25 (2010) 255–267.
- [20] W.H. Mahjoub, H. Osman, G.M. Aly, H.264 deblocking filter enhancement, in: *International Conference on Computer Engineering & Systems*, 2011, pp. 219–224.
- [21] S. Forchhammer, H. Li, J.D. Andersen, No-reference analysis of decoded MPEG images for PSNR estimation and post-processing, *Journal of Visual Communication and Image Representation* 22 (4) (2011) 313–324.
- [22] J.G. Daugman, Spatial visual channels in the Fourier plane, *Vision Research* 24 (9) (1984) 891–910.
- [23] W.B. Zhao, Z.H. Zhou, Fuzzy blocking artifacts reduction algorithm based on human visual system, in: *Proc. Int. Conf. Machine Learning and Cybernetics*, vol. 3, 2007, pp. 1626–1630.
- [24] A. Kuijper, The Deep Structure of Gaussian Scale Space Images, PhD Dissertation, Universiteit Utrecht, The Netherlands, 2002.
- [25] P. Perona, J. Malik, Scale-space and edge detection using anisotropic diffusion, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 12 (7) (1990) 629–639.
- [26] J. Weickert, H. Scharf, A scheme for coherence enhancing diffusion filtering with optimized rotation invariance, *Journal of Visual Communication and Image Representation* 13 (2002) 103–118.
- [27] H. Hassanpour, M. Nikpour, Using diffusion equations for improving performance of wavelet-based image denoising techniques, *IET Image Processing* 4 (6) (2010) 452–462.
- [28] S.K. Weeraratna, C. Kamath, PDE based nonlinear diffusion techniques for denoising scientific/industrial images: an empirical study, in: *Proc. of SPIE Electronic Imaging*, San Jose, pp. 279–290, 2002.
- [29] Z. Krivák, K. Mikula, An adaptive finite volume scheme for solving nonlinear diffusion equations in image processing, *Journal of Visual Communication and Image Representation* 13 (2002) 22–35.
- [30] G. Aubert, P. Kornprobst, *Mathematical Problems in Image Processing: Partial Differential Equations and the Calculus of Variations*, second ed., Springer, 2006.
- [31] S.T. Acton, Y. Yongjin, Speckle reducing anisotropic diffusion, *IEEE Transactions on Image Processing* 11 (11) (2002) 1260–1270.
- [32] O. Demirkaya, Smoothing impulsive noise using nonlinear diffusion filtering, in: *Computer Vision and Mathematical Methods in Medical and Biomedical Image Analysis*, Lecture Notes in Computer Science, vol. 3117, pp. 111–122, 2004.
- [33] R. Whitaker, S.M. Pizer, A multiscale approach to nonuniform diffusion, *CVGIP: Image Understanding* 57 (1) (1993) 111–120, January.
- [34] B. Fischl, E. Schwartz, Adaptive nonlocal filtering: a fast alternative to anisotropic diffusion for image enhancement, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 21 (1999) 42–49.
- [35] J. Weickert, *Anisotropic Diffusion in Image Processing*, Stuttgart, Germany, 1998.
- [36] B. Tang, G. Sapiro, V. Caselles, Color image enhancement via chromaticity diffusion, *IEEE Transactions on Image Processing* 10 (5) (2001) 701–707.
- [37] O. Demirkaya, M.H. Asyali, P.K. Sahoo, *Image Processing with MATLAB: Applications in Medicine and Biology*, CRC Press, 2008.
- [38] J. Ling, A. Bovik, Smoothing low SNR molecular images via anisotropic median-diffusion, *IEEE Transactions on Medical Imaging* 21 (4) (2002) 377–384.
- [39] J. Rajan, K. Kannan, M.R. Kaimal, An improved hybrid model for molecular image denoising, *Journal of Mathematical Imaging and Vision* 31 (1) (2008) 73–79.
- [40] M. Elad, On the origin of the bilateral filter and ways to improve it, *IEEE Transactions on Image Processing* 11 (10) (2002) 1141–1151.
- [41] Y. Nie, K. Barner, The fuzzy transformation and its application in image processing, *IEEE Transactions on Image Processing* 15 (4) (2006) 910–927.
- [42] E. Nadernejad, S. Forchhammer, J. Korhonen, Artifact reduction of compressed images and video combining adaptive fuzzy filtering and directional anisotropic diffusion, in: *Proc. European Workshop on Visual Inf. Proc. (EUVIP'11)*, Paris, France, Jul. 2011, pp. 24–29.
- [43] S. Chebbo, P. Duteux, B.P. Popescu, Adaptive Deblocking filter for DCT coded video, in: *Proc. Fourth Int. Workshop on Video Proc. and Quality Metrics for Consumer Electr. (VPMQ'09)*, Scottsdale, AZ, USA, 2009.
- [44] Y.S. Prasath, Weighted Laplacian differences based multispectral anisotropic diffusion, in: *Proc. Int. Conf. Geoscience and Remote*

Please cite this article as: E. Nadernejad, et al., Enhancing perceived quality of compressed images and video with anisotropic diffusion and fuzzy filtering, *Signal Processing-Image Communication* (2012), <http://dx.doi.org/10.1016/j.image.2012.12.001>

## ARTICLE IN PRESS

E. Nadernejad et al. / Signal Processing: Image Communication 8 (2012) 120–131

19

- Sensing Symposium, Machine Learning and Cybernetics, 2011, pp. 4042–4045.
- [45] J. Weickert, Coherence-enhancing diffusion of colour images, Proc. Seventh Nat'l Symp. Pattern Recognition and Image Analysis vol. 1 (1997).
- [46] G. Sapiro, D. Ringach, Anisotropic diffusion of multivalued images with application to color filtering, IEEE Transactions on Image Processing 5 (1996) 1582–1586.
- [47] G. Boccignone, M. Ferraro, T. Caelli, Generalized spatio-chromatic diffusion, IEEE Transactions on Pattern Analysis and Machine Intelligence 24 (10) (2002) 1298–1309.
- [48] F.D. Simone, D. Ticca, F. Dufaux, M. Ansorge, T. Ebrahimi, A.G. Tescher, A comparative study of color image compression standards using perceptually driven quality metrics, SPIE Optics and Photonics, Applications of Digital Image Processing 70730Z (2008), XXXI.
- [49] Z. Wang, A.C. Bovik, H.R. Sheikh, E.P. Simoncelli, Image quality assessment: from error visibility to structural similarity, IEEE Transactions on Image Processing 13 (4) (2004) 600–612.
- [50] ITU-T Rec. BT.500-11, Methodology for the Subjective Assessment of the Quality of Television Pictures, International Telecommunication Union, Geneva, 2004.
- [51] A. Watson, and M. Sasse, Measuring perceived quality of speech and video in multimedia conferencing applications, in: Proc. of ACM Multimedia'98, Bristol, UK, September 1998.
- [52] J.-S. Lee, F. De Simone, T. Ebrahimi, Subjective quality evaluation via paired comparison: application to scalable video coding, IEEE Transactions on Multimedia 13 (4) (2011) 882–892.
- [53] Z. Wang, A. Bovik, A universal image quality index, IEEE Signal Processing Letters 9 (3) (2002) 81–84.
- [54] <videoprocessing.ucsd.edu/~dungvo/MCSTF.html>.

Please cite this article as: E. Nadernejad, et al., Enhancing perceived quality of compressed images and video with anisotropic diffusion and fuzzy filtering, Signal Processing-Image Communication (2012), <http://dx.doi.org/10.1016/j.image.2012.12.001>

## A.5 Adaptive deblocking and deringing of H.264/AVC video sequences

E. Nadernejad, **N. Burini**, and S. Forchhammer, “Adaptive Deblocking and Deringing of H.264-AVC Video Sequences”, in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2013, pp. 2508–2512

Reference: [8]

## ADAPTIVE DEBLOCKING AND DERINGING OF H.264/AVC VIDEO SEQUENCES

Ehsan Nadernejad, Nino Burini, Søren Forchhammer

Department of Photonics Engineering, Technical University of Denmark, Ørstedts Plads,  
2800 Kgs. Lyngby, Denmark

## ABSTRACT

We present a method to reduce blocking and ringing artifacts in H.264/AVC video sequences. For deblocking, the proposed method uses a quality measure of a block based coded image to find filtering modes. Based on filtering modes, the images are segmented to three classes and a specific deblocking filter is applied to each class. Deringing is obtained by an adaptive bilateral filter; spatial and intensity spread parameters are selected adaptively using texture and edge mapping. The analysis of objective and subjective experimental results shows that the proposed algorithm is effective in deblocking and deringing low bit-rate H.264 video sequences.

**Index Terms**— deblocking, deringing, H.264, Bilateral Filter, Post-processing

## 1. INTRODUCTION

Block based video codecs like MPEG-4 [1] and H.264/AVC [2] may suffer from ringing and blocking artifacts, which require effective post-processing to be reduced. Post-processing improves image quality without changing existing standards.

Many deblocking and deringing algorithms have been proposed for compressed images and videos [3–13]. Kim [3] proposed an adaptive deblocking algorithm for low bitrate video, where the DC and AC values are used to label each block as low or high activity; then, based on the classification, two kinds of low-pass filters are adaptively applied on each block. A method combining the directional anisotropic diffusion equations with adaptive fuzzy filtering for removing blocking and ringing artifacts was presented in [4]. Zhai proposed an algorithm for deblocking [9], consisting of three parts: local AC coefficient regularization (ACR) of shifted blocks in the discrete cosine transform (DCT) domain, block-wise shape adaptive filtering (BSAF) in the spatial domain, and a quantization constraint (QC) in the DCT domain. Yao et al. [8] introduced an algorithm using histogram driven diffusion coefficients for post-processing.

This work introduces a new algorithm to reduce blocking and ringing artifacts in H.264 video sequences. Deblocking is done with a decision mode-based algorithm using local characteristics of the blocks and a quality metric of each frame (I, B, P). After deblocking, an adaptive bilateral filter is applied

to the regions with ringing artifacts. The experimental results show that the proposed algorithm effectively reduces blocking and ringing, outperforming other methods with respect to PSNR, MSSIM and subjective tests.

The rest of this paper is organized as follows. The proposed algorithm is described in Section 2. Section 3 shows the experimental results on H.264/AVC video sequences. Finally, we conclude in Section 4.

## 2. PROPOSED ALGORITHM

The proposed algorithm consists of two steps: deblocking and deringing. In the first step, the quality of each frame (I, P, B) is calculated using a quality metric and deblocking is done using decision modes. In the second step, a bilateral filter with adaptive spatial and intensity spread parameters is applied to the deblocked image for deringing.

The deblocking scheme is based on region classification with respect to the activity across block boundaries; depending on the classification, three different filtering modes are applied in the horizontal and vertical directions. Hard filtering is used on flat areas, whereas weak filtering is used to preserve sharpness in areas of high spatial or temporal activity. An intermediate mode (without filtering) is used to solve the problem of too coarse a decision and avoid either excessive blurring or inadequate removal of the blocking effect. Figure 1 presents a flowchart of the proposed algorithm.

## 2.1. Deblocking

In the deblocking step, the decision mode is done based on a pixel quality metric and predefined thresholds. Appropriate two steps filtering is then performed based on decision modes.

## 2.1.1. Quality Measure for Pixels in H.264 sequence

A compressed video sequence is mainly degraded by coarse quantization of DCT coefficients and inaccurate motion compensation. Due to different quantization steps and different frame types (I, P, B), pixels are distorted with different degrees and providing different qualities. Based on quantization step size and frame types, we estimate a quality parameter for each pixel which is used in the decision mode step.

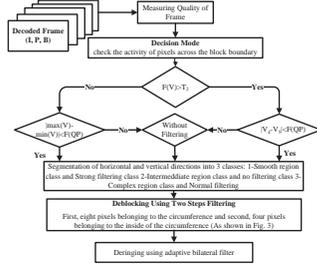


Fig. 1: Flowchart of the proposed algorithm.

The quality measure ( $Q_M$ ) is defined to reflect approximate MSE for each pixel in I, P and B frames [10]:

$$Q_M = \sqrt{12 \times MSE} \quad (1)$$

Pixels with smaller  $Q_M$  values are considered to have higher quality. This pixel quality parameter cannot reflect the quality of each individual pixel accurately, and it is just used to compare approximately individual pixels with different quantization step and frame type [10, 11].

The curves shown in Fig.2 were obtained by measuring the MSE of the luminance components of H.264/AVC decoded sequences.  $QP$  determines the quantizer step size [11]. The Laplacian distribution is used to model the MSE quality as shown in Fig.2. The results indicate that intra coded frames (I) provide the best quality, and that unidirectional prediction frame (P) have better quality than bidirectional prediction frame (B). As  $QP$  increases, degradations for I, P and B frames are all increasing, while the quality differences among I, P and B frames are decreasing. In this paper, these training data are used to describe relative comparisons between different coding modes. All the settings and testing in later experiments are based on these curves. With the  $QP$  value and frame type we can calculate the quantization step size ( $Q_s$ ) and use Fig.2 to get an MSE estimate which provided  $Q_M$  using Eq.1. The decision modes and segmentation step use the following function of  $Q_M$ :

$$F(QP) = \sqrt{Q_M} \quad (2)$$

### 2.1.2. Decision Modes and Segmentation

This step classifies the pixels activity in the regions to be filtered and applies the appropriate filter depending on the features of the region. The filtering modes are determined based

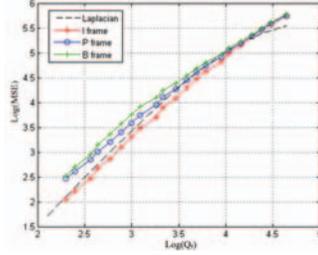


Fig. 2: MSE vs  $Q_s$ , measured on mobcal (CIF): rate control is disabled, different  $QP$  values chosen for the different points [11].

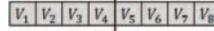


Fig. 3: Position of filtered pixels and pixel vector used for the decision mode process in horizontal direction. For vertical direction the pixel vector is the same.

on the variation of activity in vertical and horizontal pixel vectors at each  $4 \times 4$  block boundary, as shown in Fig. 3.

In this step, flat regions and complex regions are classified by local characteristics. An activity factor is assigned to the pixels inside each vector of pixels at the  $4 \times 4$  block boundaries, as described in Fig. 3. The activity is as follows:

$$R(V) = \sum_{i=1}^7 \phi(v_i - v_{i+1}), \quad (3)$$

where  $0 \leq R(V) \leq 7$  and

$$\phi(\Delta) = \begin{cases} 1, & \text{if } |\Delta| < T_1 \\ 0, & \text{otherwise} \end{cases}, \quad (4)$$

where  $T_1$  is a fixed threshold (should be set to a small value),  $V$  represents the eight-pixels vector and  $v_i$  are the pixel values. The activity factor  $R(V)$  reflects the activity in  $V$  across block boundary; it also represents the number of detected edges inside  $V$ . If the value of  $R(V)$  is smaller than a certain threshold  $T_2$ , and the difference between the maximum and minimum values of  $V$  is smaller than  $F(QP)$ , we assume  $V$  to be in a complex region and apply the filter for complex region. If  $R(V)$  is bigger than  $F(QP)$ , then it is does not need filtering. If  $R(V) > T_2$ , the two pixel values on either side of the block boundary ( $v_1$  and  $v_8$ ) are considered. If the absolute

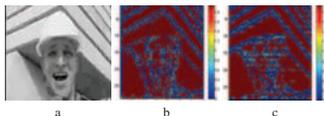


Fig. 4: Decision mode of the 25<sup>th</sup> frame of the Foreman sequences, a) coded frame, b) horizontal direction modes, c) vertical direction modes.

difference of two pixels is smaller than  $F(QP)$ , we assume  $V$  to be in a smooth region, otherwise it does not need filtering. In this work,  $T_1 = 6$  and  $T_2 = 2$ .

Based on the decisions made in horizontal and vertical directions, the frame is segmented in three no filtering (N), weak filtering (W) and hard filtering (H) regions. Figure 4 shows an example of segmentation.

### 2.1.3. Two steps filtering for deblocking

Two steps filtering is done after segmentation and labeling of each pixel. A  $6 \times 6$  filtering window is centered at the intersection of four  $4 \times 4$  pixel blocks as shown in Fig. 5. The filtering window is placed at the upper left corner of the frame and is shifted across the whole frame.

Deblocking is done in two steps. In the first step, only eight pixels are filtered at the intersection of four  $4 \times 4$  pixel blocks ( $x_1 \dots x_8$ ). As mentioned before, there are two options for each pixel in the both vertical and horizontal directions. After segmentation, if no filtering mode is selected in any direction with other filtering modes, only one dimensional filters are required. For instance, in NW or WN modes just apply a weak 1D filter on the target pixels in vertical and horizontal directions, respectively. If NH or HN is selected then a hard 1D filter is applied to the target pixel in one of both directions. When the filtering mode belongs to the weak filtering and hard filtering mode (WH, HW, HH), 2D filtering is applied on the pixel. Equation 5 shows the updated values of the  $x_{14}$  in different modes ( $\gg$  is the bitshift operator).

$$x_{14} = \begin{cases} y_1 + 5x_1 + 3x_2 - y_2 \gg 3 & \text{if NW,} \\ 2a_7 + y_1 + 2x_1 + x_2 + y_2 + a_8 \gg 3 & \text{if NH,} \\ 4(x_1 + a_3) + 2(x_2 + y_2 + y_3) + y_1 + y_2 \gg 4 & \text{if WH,} \\ 4(a_2 + a_3 + a_4 + x_1) + 2(y_1 + x_5 + y_4 \\ + y_6) + a_7 + x_2 + y_2 + a_8 + y_3 + y_5 + \\ x_3 + x_6 \gg 5 & \text{if HH.} \end{cases} \quad (5)$$

The other pixels are filtered in the same way. To limit computation, the weighting matrix of the 2D filter is simplified and some coefficients are cut or rounded. The literature includes different methods for simplification [3, 5].

$a_6$	$a_2$	$a_3$	$a_4$	$a_5$	$a_6$
$a_7$	$y_1$	$x_1$	$x_2$	$y_2$	$a_8$
$a_9$	$x_5$	$y_2$	$y_4$	$x_7$	$a_{10}$
$a_{11}$	$x_6$	$y_2$	$y_6$	$x_8$	$a_{12}$
$a_{13}$	$y_7$	$x_3$	$x_4$	$y_8$	$a_{14}$
$a_{15}$	$a_{16}$	$a_{17}$	$a_{18}$	$a_{19}$	$a_{20}$

Fig. 5: Pixels must be filtered in two steps filtering in window ( $6 \times 6$ ).

At the end of the first step, pixels belonging to  $x_1 \dots x_8$  are filtered and in the second step, the remaining pixels belonging to  $y_3 \dots y_6$  are filtered by applying the appropriate filter according to the pre-assigned filtering mode. The pixels will update according to their filtering mode as follows:

$$y_{3n} = \begin{cases} 3x_5 + y_3 + y_7 - y_4 \gg 2 & \text{if NH} \\ 2x_5 + 5y_3 + 3y_4 - 2x_6 \gg 3 & \text{if NW} \\ 6x_3 + 4y_3 + 2y_4 + 4x_7 + 2y_5 - x_1 - x_3 \gg 4 & \text{if HW} \\ x_5 - x_7 + x_1 - x_3 + 4y_3 + 2y_5 + 2y_4 \gg 3 & \text{if WW} \\ 2(x_5 + x_7 + x_1 + x_3) + y_5 + 6y_3 + y_4 \gg 4 & \text{if HH} \end{cases} \quad (6)$$

For symmetric filtering modes, the filtering values are simply computed in a symmetric manner.

### 2.2. Bilateral Filter for Deringing

After removing the blocking artifacts from the frame, an adaptive bilateral filter is used to remove ringing artifacts. The bilateral filter is a nonlinear weighted averaging filter, obtained by combining two Gaussian filters; one filter works in spatial domain, other filter works in intensity domain [14]. The weights depend on both the spatial distance and the intensity distance with respect to the center pixel. The main feature of the bilateral filter is its ability to preserve edges while doing spatial smoothing. At pixel location  $x$ , the output of a bilateral filter can be formulated as follows:

$$J(x) = \frac{1}{Z} \sum_{y \in \psi(x)} e^{-\frac{\|y-x\|^2}{2\sigma_s^2}} e^{-\frac{\|I(y)-I(x)\|^2}{2\sigma_r^2}}, \quad (7)$$

where  $\sigma_s$  and  $\sigma_r$  are parameters controlling the fall-off of weight in spatial and intensity domains, respectively.  $\psi(x)$  is the spatial neighborhood of pixel  $I(x)$  and  $Z$  is a normalization constant:

$$Z = \sum_{y \in \psi(x)} e^{-\frac{\|y-x\|^2}{2\sigma_s^2}} e^{-\frac{\|I(y)-I(x)\|^2}{2\sigma_r^2}}. \quad (8)$$

The behavior of the bilateral filter is determined by  $\sigma_s$  and  $\sigma_r$ . For deringing, these parameters should be chosen carefully, since it is desirable to avoid over-smoothing texture regions and to preserve edges in edge regions. These could be done first by estimating the texture regions and discontinuity of the

edges, and then control the extent of smoothing and sharpening through the  $\sigma_d$  and  $\sigma_r$  values. In the proposed method, each  $4 \times 4$  block is classified into one of the four categories: strong edge, weak edge, texture and smooth blocks. For a smooth region, the value of the  $\sigma_d$  can be large, otherwise it should be small. Classification is done by computing the standard deviation (STD) in a  $4 \times 4$  window around each pixel and comparing the maximum STD in each  $4 \times 4$  block with a set of predetermined thresholds as follows:

$$\sigma_d = \begin{cases} \text{StrongEdge}, \sigma_d = 0.8 & \text{if MaxSTD} \in [35, \infty) \\ \text{WeakEdge}, \sigma_d = 1.8 & \text{if MaxSTD} \in [25, 35) \\ \text{Texture}, \sigma_d = 2.8 & \text{if MaxSTD} \in [15, 25) \\ \text{Smooth}, \sigma_d = 3.8 & \text{if MaxSTD} \in [0, 15) \end{cases} \quad (9)$$

The optimal  $\sigma_r$  value of the bilateral filter is linearly proportional to the standard deviation of the noise ( $\sigma_r = \alpha \times \sigma_n$ ). The noise variance is estimated with the robust median noise estimator technique [11]. In the proposed algorithm, the value of  $\alpha$  is set to  $1/3$  in each  $4 \times 4$  block. The calculation of  $\sigma_d$  and  $\sigma_r$  are repeated for all blocks to obtain the block spatial map  $M_{\sigma_d}$  and the block intensity map  $M_{\sigma_r}$ .

### 3. SIMULATION RESULTS

The performance of the proposed algorithm was evaluated on H.264/AVC video sequences through comparison with our implementation of several state-of-the-art spatial post-processing algorithms [3–9, 12, 13]<sup>1</sup>. The GOP structure was defined as  $(IPPB)_{12}$ . Two different types of experiments have been performed. In the first experiment, the algorithm was applied with two different quantization parameters ( $QP = 35, 45$ ) with the in-loop deblocking filter enabled. In the second experiment the in-loop deblocking filter was disabled. Several CIF (4:2:0) test sequences were chosen: Akiyo, Bus, Coastguard, Container, Cycling, Foreman, Hall, Mobcal, Mother and Daughter. The algorithms were applied on the first 100 frames of each sequence. The qualities of the different algorithms have been compared in terms of Weighted-PSNR and Weighted-MSSIM, where the luma and chroma components have a weight of  $2/3$  and  $1/6$ , respectively [15, 16]. The comparative objective results for the first experiment are summarized in Table 1. It can be seen that the proposed algorithms achieves higher PSNR and MSSIM compared to the other algorithms.

In the second experiment, the H.264/AVC in-loop deblocking filter was disabled. The proposed algorithm reaches higher PSNR and MSSIM when compared to the in-loop filter alone. Table 2 shows the performance of the proposed algorithm against H.264/AVC when the in-loop filtering is disabled on Akiyo video sequences.

Figure 6 visually compares the in-loop filter and the proposed post-processing algorithm for deblocking and deringing. It can be seen that the blocking and ringing artifacts are

<sup>1</sup>The software for [9] was provided by Zhai

**Table 1:** The average results of post-processing H.264/AVC video test sequences using different algorithms.

Metric	PSNR		MSSIM	
	35	45	35	45
QP	35	45	35	45
H.264/AVC	34.76	30.51	0.906	0.810
[3]	34.09	30.35	0.898	0.806
[4]	35.09	30.63	0.911	0.812
[5]	35.03	30.52	0.910	0.811
[6]	34.66	30.49	0.901	0.809
[7]	34.82	30.57	0.907	0.813
[9]	35.03	30.59	0.910	0.813
[8]	35.00	30.36	0.907	0.809
[12]	35.04	30.39	0.909	0.810
[13]	34.82	30.44	0.908	0.809
Proposed	35.18	30.62	0.911	0.814

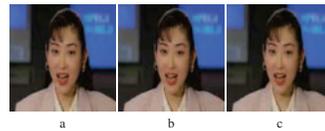
**Table 2:** Results of H.264/AVC video when the in-loop filtering is disabled/enabled and with the proposed algorithm on Akiyo video.

Metric	PSNR		MSSIM	
	35	45	35	45
QP	35	45	35	45
Disabled in-loop	33.15	28.21	0.895	0.793
Enabled in-loop	33.25	28.36	0.912	0.806
Proposed	33.30	28.43	0.912	0.812

more effectively attenuated in both images, resulting in a better perceptual quality for the reconstructed video.

### 4. CONCLUSION

We have proposed an adaptive post-processing algorithm for blocking and ringing artifact reduction in H.264/AVC video sequences. The algorithm uses a quantization parameter to estimate the quality of each frame. Deblocking is performed using a quality metric and the activity of pixels across of the block boundary; a deringing algorithm is applied to the areas which have ringing artifacts using an adaptive bilateral filter. Results show that the proposed algorithm improves the objective and subjective quality of H.264 video sequences.



**Fig. 6:** The comparison of filter result on Akiyo (75<sup>th</sup> frame). a) Compressed frame, b) In-loop filter, c) Proposed algorithm.

## 5. REFERENCES

- [1] "MPEG-4 Video Verification Model Version 8.0," MPEG Video Group, ISO/IEC JTC1/SC29/WG11 N1796, July 1997.
- [2] "Draft ITU-T Recommendation and Final Draft International Standard Joint Video Specification," Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG, ITU-T Rec. H.264-ISO/IEC, 14496-10 AVC/JVT G050, 2003.
- [3] Changick Kim, "Adaptive post-filtering for reducing blocking and ringing artifacts in low bit-rate video coding," *Sig. Proc.: Image Comm.*, vol. 17, no. 7, pp. 525–535, 2002.
- [4] Ehsan Nadernejad, Sren Forchhammer, and Jari Korhonen, "Artifact reduction of compressed images and video combining adaptive fuzzy filtering and directional anisotropic diffusion," in *3rd European Workshop on Visual Information Processing (EUVIP), Paris, July 4-6, 2011*, pp. 24–29.
- [5] Salim Chebbo, Philippe Durieux, and Batrice Pesquet-Popescu, "Adaptive Deblocking Filter for DCT Coded Video," in *Proc. Fourth Int. Workshop on Video Process. and Quality Metrics for Consumer Electr.*, Jan 2009.
- [6] Shen-Chuan Tai, Yen-Yu Chen, and Shin-Feng Sheu, "Deblocking filter for low bit rate MPEG-4 video," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 6, pp. 733 – 741, June 2005.
- [7] Hao-Song Kong, Yao Nie, A. Vetro, Huifang Sun, and K.E. Barner, "Adaptive fuzzy post-filtering for highly compressed video," in *Image Processing, 2004. ICIP '04. 2004 International Conference on*, oct. 2004, vol. 3, pp. 1803 – 1806 Vol. 3.
- [8] Susu Yao, Keng Pang Lim, Xiao Lin, and S. Rahardja, "A post-processing algorithm using histogram-driven anisotropic diffusion," in *Circuits and Systems, 2005. ISCAS 2005. IEEE International Symposium on*, may 2005, pp. 4233 – 4236 Vol. 5.
- [9] Guangtao Zhai, Wenjun Zhang, Xiaokang Yang, Weisi Lin, and Yi Xu, "Efficient Deblocking With Coefficient Regularization, Shape-Adaptive Filtering, and Quantization Constraint," *IEEE Trans. Multimedia*, vol. 10, no. 5, pp. 735 –745, Aug 2008.
- [10] Bo Martins and Søren Forchhammer, "A unified approach to restoration, deinterlacing and resolution enhancement in decoding MPEG-2 video," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, no. 9, pp. 803–811, 2002.
- [11] Xing Huang, Huiying Li, and Søren Forchhammer, "A Multi-Frame Post-Processing Approach to Improved Decoding of H.264/AVC Video," in *Proc. IEEE ICIP*, 2007, vol. 4, pp. 381–384.
- [12] Ming Zhang and Bahadır K. Gunturk, "Compression artifact reduction with adaptive bilateral filtering," in *Proc. SPIE*, 2009, vol. 7257, pp. 72571A–72571A–11.
- [13] Sung Deuk Kim, Jaeyoun Yi, Hyun Mun Kim, and Jong Beom Ra, "A deblocking filter with two separate modes in block-based video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, no. 1, pp. 156 –160, Feb 1999.
- [14] Ming Zhang and Bahadır K. Gunturk, "Multiresolution Bilateral Filtering for Image Denoising," *IEEE Trans. Image Process.*, vol. 17, no. 12, pp. 2324 –2333, Dec 2008.
- [15] Francesca De Simone, Daniele Ticca, Frederic Dufaux, Michael Ansonge, and Touradj Ebrahimi, "A comparative study of color image compression standards using perceptually driven quality metrics," in *Proc. SPIE Optics and Photonics, San Diego, CA USA, August 11-14, 2008*.
- [16] Ehsan Nadernejad, Jari Korhonen, Søren Forchhammer, and Nino Burini, "Enhancing perceived quality of compressed images and video with anisotropic diffusion and fuzzy filtering," *Signal Processing: Image Communication*, vol. 28, no. 3, pp. 222–240, 2013.

## A.6 Optimal backlight scanning for 3D crosstalk reduction in LCD TV

**N. Burini**, X. Shu, L. Jiao, S. Forchhammer, and X. Wu, “Optimal Backlight Scanning for 3D Crosstalk Reduction in LCD TV”, in *IEEE International Conference on Multimedia & Expo (ICME)*, To be presented, 2013

Reference: [9]

### OPTIMAL BACKLIGHT SCANNING FOR 3D CROSSTALK REDUCTION IN LCD TV

Nino Burini<sup>1</sup>, Xiao Shu<sup>2</sup>, Liangbao Jiao<sup>3</sup>, Søren Forchhammer<sup>1</sup>, Xiaolin Wu<sup>2</sup>

<sup>1</sup> Department of Photonics Engineering, Technical University of Denmark, Kongens Lyngby, Denmark

<sup>2</sup> Department of Electrical and Computer Engineering, McMaster University, Hamilton, ON, Canada

<sup>3</sup> School of Communications Engineering, Nanjing Institute of Technology, Nanjing, P. R. China

#### ABSTRACT

This work presents a method to determine the optimal backlight scanning signals to minimize crosstalk for time-sequential stereoscopic 3D on LCD TV with active shutter glasses. The solution is obtained through optimization of the variables defined by a model of backlight scanning that considers important aspects like liquid crystal transitions and light diffusion, subject to constraints that ensure the rendition of a uniform backlight. Compared with basic backlight scanning, the proposed method can increase luminance at a given crosstalk level or reduce crosstalk at a given luminance level.

**Index Terms**— Optimization, Liquid crystal displays, Time-sequential stereo 3D, 3D crosstalk, Backlight, Scanning, Strobing, Blinking, Three dimensional TV

#### 1. INTRODUCTION

Among display technologies, liquid crystal displays (LCDs) are today the most widespread on the TV market. Traditionally used to visualize two dimensional (2D) content, LCDs can also be used to display stereo three dimensional (3D) content [1]. The effect of stereo vision is achieved by showing two different views of the same scene to the left and the right eye; this is possible with either passive glasses or active glasses. In the first case, left and right frame are displayed at the same time but on different partitions of the screen (i.e. even and odd lines); the partitions emit light with different polarization and the use of glasses with properly polarized filters allows the right frame to reach only the right eye and the left frame to reach only the left eye. In the second case, the left and the right frames alternate on the screen in sequence; the frames are delivered to the intended eye with the use of active shutter glasses based on liquid crystals (LCs), which become clear to let only the intended frame through and opaque to block the other. To maintain the input frame rate, the display refresh rate must be double (i.e. a 60Hz video requires a 120Hz refresh rate). This second approach is called *time-sequential stereoscopic visualization* and is the case considered in this work.

One of the most important aspects affecting 3D video quality is the presence and intensity of *crosstalk*. Crosstalk

is the incomplete isolation of the left and right image channels so that one image leaks into the other; the term is often confused with *ghosting*, which is the perception of crosstalk [2]. When using shutter glasses with LCDs, crosstalk is due to several factors, including the quality of the shutter glasses, their synchronization with the display, the viewing angle, the response time of the display pixels and the image update method of the screen [1]. The last two aspects depend on the display and are addressed in this work. In LCDs, frames are updated with a progressive scanning process where pixels are addressed in sequence, typically starting from the top-left pixel and then proceeding left-to-right and top-to-bottom. After being addressed, each pixel completes the transition to its new state and becomes stable after a finite time, called *response time*. These two facts imply that, during the update process, there is more than one frame shown on the display; depending on the characteristics of the display, it can happen that there is never one single frame shown on the screen. This is one of the causes of crosstalk independent from other factors like those due to the shutter glasses.

In this work, we focus on reducing crosstalk using LCD *backlight*. The backlight is the component of an LCD that emits light; this light is modulated by LC based pixels, which give it the desired intensity and color, depending on the input image. Traditionally, backlight used to be global, that is with a uniform luminance across the whole display, and always turned on. More recently, new screens with *local backlight* have been introduced: here, the backlight is divided in several sectors or segments, each one with a light source (usually light emitting diodes, or LEDs) that can be controlled independently. Local backlights have several advantages, for example to allow dynamic backlight dimming, where the backlight adapts dynamically to the displayed image to reduce power consumption and improve contrast. Local backlights can be also used to reduce crosstalk in time-sequential stereo visualization, through the techniques of *backlight strobing* (also called blinking [3]) and *backlight scanning*. In backlight strobing the whole backlight is turned on only when all the pixels are stable, otherwise is turned off; this way, each frame is shown only when all its pixels have been updated completely, assuring that there is no residue of the previous one; unfortunately this might not always be possible if the update

process is too long. Backlight scanning is instead done during the update process: the backlight segments are turned on and off to follow the LC scanning and to highlight the stable pixels against the others. Backlight strobing and scanning can be combined, so that the first is done while the pixels are updated and the second is done when they are all stable. It should be noted that luminance is a scarce resource when dealing with 3D TV on LCDs with passive glasses, because of the many elements that dissipate the backlight (display light guides, light diffusers, light polarizers, LCs, color filters and glasses LCs). Both strobing and scanning reduce light emission, so it is important to maximize it with acceptable crosstalk levels.

While backlight strobing is overall simple and only requires to know when all pixels are stable, backlight scanning is a more complex problem. A very simple way to do backlight scanning is to turn on one segment at a time, in sequence, to follow the pixel scanning. This approach has been used in [4] and we refer to it as basic backlight scanning. Although simple, basic backlight scanning ignores display characteristics and, with a large number of segments, can reduce luminance dramatically. In this work, we propose a model for backlight scanning that we use to minimize crosstalk at a given luminance level. The model includes important aspects like light diffusion in the backlight and the pixel response for display specific optimization.

The rest of this paper is structured as follows. Section 2 presents the model in all its aspects. The model is used to formulate an optimization problem in Section 3. Experiments based on the model and results are reported in Section 4. Finally, conclusions are drawn in Section 5.

## 2. MODELING

In this section we define a model for backlight scanning including its main aspects. Based on the model, we propose a solution to find the optimal backlight scanning strategy minimizing 3D crosstalk. To avoid issues like changing luminance levels, the model is defined image-independent.

### 2.1. Timing of scanning and strobing

We consider the case of LCD screens which start drawing a new frame from the top-left pixel, going left-to-right and top-to-bottom. The *frame time*  $t_f$  is the time between the start of two consecutive frames. On a display with a refresh rate of 120Hz,  $t_f = \frac{1}{120}$  s. The pixels are addressed one by one at time  $t_i$ ; the time required to address all of them is called *addressing time* ( $t_a$ ). After being addressed, each pixel completes the transition after the *response time*  $t_r$ .

We define *scanning time* ( $t_s$ ) as the time during  $t_f$  when backlight scanning is performed:

$$t_s = \begin{cases} t_a + t_r & \text{if } t_a + t_r < t_f \\ t_f & \text{if } t_a + t_r \geq t_f \end{cases} \quad (1)$$

By definition,  $t_s$  cannot be longer than  $t_f$ . When all the pixels are stable there is no need to scan the backlight and strobing can be done by turning all segments on. Strobing lasts for a  $t_f - t_s$  long time, therefore it is not possible if  $t_s = t_f$ , because there is never an instant when all pixels are stable. Figure 1 includes a visualization of these concepts. In general, during  $t_s$  at least two frames are partially shown on the screen; depending on  $t_a$  and  $t_r$ , the frames can be three or more. In this work we assume  $t_a < t_f$ , which guarantees that the frames are at most two, and  $t_s < t_f$ , which ensures that each pixel becomes stable before being addressed again.

During one loop of backlight scanning loop lasting  $t_s$ , the backlight is updated  $Q$  times. The updates happen at regular intervals, at time  $t_k$  where  $k = 0, \dots, Q - 1$ , so that

$$t_{k+1} = t_k + \frac{t_s}{Q} \quad (2)$$

During the time interval  $[t_k, t_{k+1})$ , or equivalently during  $[t_k, t_k + \frac{t_s}{Q})$ , the backlight is fixed. For simplicity, we assume that  $t_0$  coincides with the beginning of the new frame, that is when the top-left pixel starts the transition.

### 2.2. Backlight

Local backlights have several independently controllable light sources, usually LEDs, assigned to specific areas of the display, called segments. In the following discussion we use the terms LEDs and segments interchangeably.

Each LED contributes to the luminance of many pixels, and the luminance of each pixel is determined by many LEDs. The interaction between LEDs and pixels is described by the point spreading functions (PSFs), which specify the influence of each LED on each pixel.

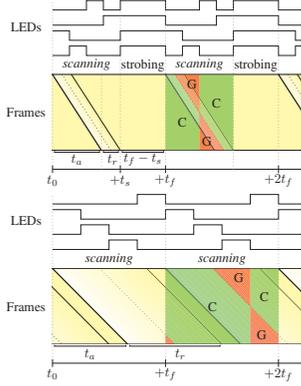
LED  $j$  is controlled by the time varying signal  $r_j(t)$ , the values of which range between 0 and 1, 0 meaning that the LED is off and 1 meaning that it is at full power. The luminance emitted by LED  $j$  is proportional to  $r_j(t)$ . The backlight luminance  $b_i(t)$  emitted by all  $M$  LEDs at pixel  $i$  is

$$b_i(t) = \sum_{j=1}^M h_{ij} \cdot r_j(t) \quad (3)$$

where  $h_{ij}$  is the contribution of LED  $j$  to pixel  $i$  when the LED is at full power. As  $r_j(t)$ ,  $b_i(t)$  is valued between 0 and 1. The complete set of values  $h_{ij}$  for all LEDs and all pixels defines the PSFs of the backlight. We assume uniform backlight, therefore all  $h_{ij}$  for pixel  $i$  sum up to 1 and consequently all  $h_{ij}$  sum up to the number of pixels, defined as  $N$ .

During  $[t_k, t_k + \frac{t_s}{Q})$ , the  $r_j(t)$  values do not change, as well as  $b_i(t)$ . We indicate the backlight luminance for pixel  $i$  and the value of the LED  $j$  during this time as  $r_{jk}$  and  $b_{ik}$ , respectively:

$$b_{ik} = \sum_{j=1}^M h_{ij} \cdot r_{jk} \quad (4)$$



**Fig. 1.** Backlight scanning/strobing timeline for  $t_s < t_f$  with short  $t_n$  and  $t_r$  (top) and for  $t_s = t_f$  with larger  $t_n$  and  $t_r$  (bottom); in the latter case strobing does not occur; the vertical axis of Frames represent vertical screen position; the shading indicates pixel transition to the new frame; the red and green overlays show when the frames are considered correct (C) or ghost (G); the square waves are the LED control signals.

### 2.3. Frames

We introduce the definition of *correct frame* and *ghost frame* (so named because it causes ghosting): the former is the frame that is being shown on more than  $\frac{N}{2}$  pixels, while the latter is the frame shown on the remaining pixels.

In the case of two frames on the screen, there is a relation between previous/current frame and correct/ghost frame. When the current frame starts to be updated at  $t_0$ , the previous frame is the correct and the current is the ghost. When half of the pixels have changed to the current frame, this becomes the correct frame and therefore the previous becomes the ghost frame. The rigorous definition of the moment when the change happens depends on the pixel transition function (see next Section). For simplicity, in this work we define this instant to be  $t_0 + \frac{t_n + t_r}{2}$ . Figure 1 includes an illustration of these concepts.

### 2.4. Pixel transition and state

In a LCD, the luminance emitted by pixel  $i$  depends on the intensity of the backlight behind it and by its transmittance. The proposed model is image-independent, therefore we ignore transmittance. However, for backlight scanning/strobing it is necessary to know the state of the pixel in its transition from one frame to the other. We express this transition with the function  $f(t)$ , which is characterized by the response time  $t_r$ , which depends on the LC characteristics and on the initial and final gray values. To keep the model image-independent, we consider only one response time which is representative of all the gray-to-gray transitions. This could be, for example, the largest response time of a large selection of gray-to-gray transitions. The function  $f(t)$  is valued between 0 and 1 and is increasing in  $[0, t_r]$ ; its value is zero for negative  $t$  and 1 for  $t > t_r$ . The LC transition has been studied and modeled in many previous works [5, 6, 7]. For simplicity, here we use a linear transition from 0 to 1 in  $[0, t_r]$ .

With regards to the correct and ghost frame, let us express the *state* of pixel  $i$  with the periodical function  $s_i(t)$ , which is based on  $f(t)$  and valued between 0 and 1. When  $s_i(t)$  is 1, the pixel is part of the correct frame; when  $s_i(t)$  is 0, the pixel is part of the ghost frame; other values mean that the pixel is in transition from one frame to the other. The period of  $s_i(t)$  is  $t_f$  since the pixels are refreshed over the same period. The complete definition of  $s_i(t)$  is

$$s_i(t) = \begin{cases} 1 - f((t - t_i) \bmod t_f) & \text{if } t \in [t_i, \frac{t_n + t_r}{2}] \\ f((t - t_i) \bmod t_f) & \text{or if } t \in [\frac{t_n + t_r}{2}, t_i] \\ f((t - t_i) \bmod t_f) & \text{otherwise} \end{cases} \quad (5)$$

During the time interval  $[t_i, \frac{t_n + t_r}{2}]$  (or  $[\frac{t_n + t_r}{2}, t_i]$  if  $t_i > \frac{t_n + t_r}{2}$ ), the value of  $f(t)$  is inverted to reflect the fact that the transition is from the correct frame to the ghost frame. Figure 2 shows examples of  $s_i(t)$ .

We also define  $s_{ik}$  as the average state of pixel  $i$  during the time interval  $[t_k, t_k + \frac{t_f}{Q}]$ , also depicted in Figure 3:

$$s_{ik} = \frac{Q}{t_n} \int_{t_k}^{t_k + \frac{t_f}{Q}} s_i(t) dt. \quad (6)$$

### 2.5. Luminance

We introduce  $l_{C,i}(t)$  and  $l_{G,i}(t)$  as the contribution of pixel  $i$  to the luminance of the correct frame and the ghost frame:

$$l_{C,i}(t) = s_i(t) b_i(t), \quad (7)$$

$$l_{G,i}(t) = (1 - s_i(t)) b_i(t). \quad (8)$$

Equation 8 reflects the fact that the light not going through the correct frame goes through the ghost frame. The average  $l_{C,i}$

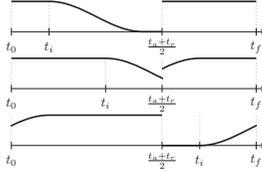


Fig. 2. Three examples of  $s_i(t)$  with different  $t_i$ .

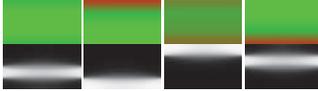


Fig. 3. Example of average pixel state (top) and backlight diffusion (bottom) during scanning time with 4 segments and 4 updates; green color corresponds to an average state mostly in the correct frame, while red corresponds to the ghost frame.

during  $[t_0, t_0 + t_s)$  can be calculated by integration:

$$l_{C,i} = \frac{1}{t_s} \int_{t_0}^{t_0+t_s} s_i(t) b_i(t) dt. \quad (9)$$

Reminding that during  $[t_0, t_0 + t_s)$  the backlight is updated  $Q$  times, that it is constant during  $[t_k, t_k + \frac{t_s}{Q})$  with  $k = 0, 1, \dots, Q-1$ , and combining Eq. 9 with Eqs. 4 and 6 we obtain

$$l_{C,i} = \frac{1}{Q} \sum_{k=0}^{Q-1} b_{ik} s_{ik}. \quad (10)$$

Assuming a uniform backlight, we can express the correct frame luminance  $L_C$  as the average of the pixel luminances:

$$L_C = \frac{1}{N} \sum_{i=1}^N l_{C,i}. \quad (11)$$

Combining Eq. 11 with Eqs. 10 and 4 gives

$$L_C = \frac{1}{NQ} \sum_{i=1}^N \sum_{k=0}^{Q-1} \sum_{j=1}^M h_{ij} s_{ik} r_{jk}. \quad (12)$$

Upon definition of the coefficients  $g_{jk} = \sum_{i=1}^N h_{ij} s_{ik}$ , Eq. 12 can be rewritten as

$$L_C = \frac{1}{NQ} \sum_{j=1}^M \sum_{k=0}^{Q-1} g_{jk} r_{jk}. \quad (13)$$

As Eq. 13 shows,  $L_C$  is simply a linear combination of the LED values  $r_{jk}$  with coefficients  $g_{jk}$ .

The luminance of the ghost frame  $L_G$  can be calculated similarly using  $l_{G,i}$  instead of  $l_{C,i}$ . It can be calculated that

$$L_G = \frac{1}{NQ} \sum_{i=1}^N \sum_{k=0}^{Q-1} b_{ik} - L_C, \quad (14)$$

which shows again that the backlight flows through either the correct or the ghost frame. We define the frame luminance  $L$ :

$$L = \frac{1}{NQ} \sum_{i=1}^N \sum_{k=0}^{Q-1} b_{ik}, \quad (15)$$

from which follows that

$$L = L_G + L_C. \quad (16)$$

### 3. OPTIMAL BACKLIGHT SCANNING

Based on the proposed model, we formulate an optimization problem to minimize crosstalk at a fixed luminance level using backlight scanning.

#### 3.1. Crosstalk

The metric used in [4] measures crosstalk between black and white frames and follows international standards (Section 17.1, Eq. 27 of [8]). However, this metric includes the contribution to crosstalk of shutter glasses and is designed for measurements with a light-meter. In order to measure crosstalk as a function of the display only and to avoid image dependency, for our model we define it as

$$\text{Crosstalk} = \frac{L_G}{L_C}, \quad (17)$$

which combined with Eq. 16 gives

$$\text{Crosstalk} = \frac{L}{L_C} - 1. \quad (18)$$

Minimizing Eq. 18 is equivalent to minimizing Eq. 17 and, for fixed  $L$ , it is equivalent to maximizing  $L_C$ .

#### 3.2. Constraints

It is necessary to enforce a constraint on  $r_{jk}$  ensuring a uniform backlight. We assume that the  $Q$  updates of the backlight during  $t_s$  are fast enough to avoid visible flickering and to produce a sensation of constant backlight. For this, it is necessary that the average value of LED  $j$  during the time interval  $[t_0, t_s)$  is constant and equal for all LEDs:

$$\frac{1}{Q} \sum_{k=0}^{Q-1} r_{jk} = d \quad j = 1, \dots, M. \quad (19)$$

The constant  $d$  must be valued within the 0 and 1. Remembering that the sum of all the  $h_{ij}$  elements is  $N$  (see Section 2.2) and combining Eqs. 4, 15 and 19, we obtain

$$L = d, \quad (20)$$

which means that the frame luminance is equivalent to the average LED value during scanning time.

### 3.3. Optimization problem

The goal of optimization is the minimization of crosstalk. For fixed  $d$ , this is equivalent to the maximization of  $L_C$ . The variables of the problem are the  $M \times Q$  LED values  $r_{jk}$ . The constraints are the uniformity constraint (Eq. 19) and the valid values of  $r_{jk}$ . The problem is therefore:

$$\begin{aligned} & \text{maximize} && \sum_{j=1}^M \sum_{k=0}^{Q-1} g_{jk} r_{jk} \\ & \text{subject to} && \sum_{k=0}^{Q-1} r_{jk} = Qd \quad j = 1, \dots, M \\ & && 0 \leq r_{jk} \leq 1 \end{aligned} \quad (21)$$

Considering that realistic maximum values of  $M$  and  $Q$  are in the order of tens and thousands, respectively, the number of variables of the problem is in the order of tens of thousands. Considering also that the value to maximize is a linear combination of variables, the complexity of the problem can be considered low. It might be more computationally demanding to calculate the  $g_{jk}$  coefficients, depending on them being determined numerically or analytically and by the modeling of  $f(t)$ . Nevertheless, the optimization problem is display specific and needs to be solved only once: after the optimal scanning control signals are known, they can be used on the backlight of the display without any additional cost.

## 4. EXPERIMENTS AND RESULTS

This section reports the results of some experiments that have been designed on the model presented above. The proposed approach for optimal backlight scanning is compared against basic backlight scanning, where the segments are turned on at full power ( $r_{jk} = 1$ ) in sequence and only one at a time. This means that the constraint given by Eq. 19 is respected and that  $d = \frac{1}{M}$ , which is also the frame luminance  $L$ .

We have simulated a display based on an existing Full HD edge-lit screen with 16 backlight segments placed in 8 rows and 2 columns. Grouping segments together enables the emulation of a coarser backlight (i.e. 4 rows and 1 column). We consider the cases of 2, 4 and 8 rows and 1 column and additionally that of global backlight where all  $r_{jk}$  are set to  $d$ .

The proposed approach was implemented in Matlab using the CVX package (version 1.21). The frame time  $t_f$  was set to  $\frac{1}{120}$  s = 8.33ms. The addressing time  $t_a$  was set to  $0.75 \times t_f$ . Three response times  $t_r$  were considered: 0ms (ideal, instantaneous response), 2ms and 4ms. The pixel response  $f(t)$  was modeled with a linear function, for simplicity. The number

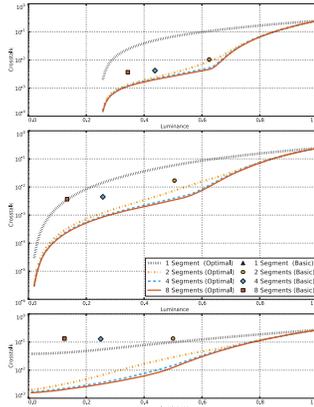
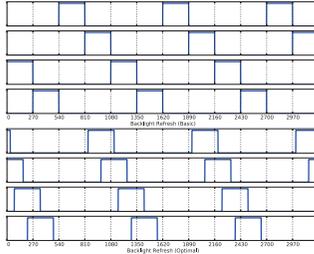


Fig. 4. Experiment with  $t_r = 0$ ms (top),  $t_r = 2$ ms (middle) and  $t_r = 4$ ms (bottom).

of backlight updates  $Q$  was set equal to the number of pixel lines (1080); this is justified by the very fast response time of LEDs, which is in the order of few microseconds or less. The minimum crosstalk was calculated for different values of  $d$  ranging from 0 to 1 with a step of 0.01.

Figure 4 shows the results for  $t_r = 0$ ms,  $t_r = 2$ ms and  $t_r = 4$ ms. The plots account for backlight scrolling and strobing combined. Since while strobing the backlight is fully turned on and all pixels are stable, average luminance over  $t_f$  is equal to  $\frac{t_a}{t_f} L_C + \frac{t_f - t_a}{t_f} L$ , while average crosstalk is equal to  $\frac{t_a}{t_f} \frac{L_C}{L_C}$ , as there is no crosstalk while strobing.

It can be seen that optimal backlight scanning improves the tradeoff between luminance and crosstalk compared to basic scanning. For instance, with  $t_r = 4$ ms and 2 segments, basic scanning has 50% luminance and  $1.335 \times 10^{-1}$  crosstalk. Optimal scanning can either reduce crosstalk to  $2.895 \times 10^{-2}$  at the same luminance level or increase luminance to 83% with the same crosstalk. The relative improvement is even larger for 4 and 8 segments. In all cases, having more backlight segments allows larger crosstalk reduction at the same luminance level. This is expected as, in these experiments, segments are grouped together to simulate coarser backlights, therefore limiting the degrees of freedom of the optimization.



**Fig. 5.** LED control signals before (top, basic backlight scanning) and after optimization (bottom) with 4 backlight segments.

It can also be seen that the size of the improvement varies with the luminance level, and that the improvement from 2 to 4 segments is rather large if compared to the 4 to 8 improvement. This might suggest that 4 backlight segments could be “good enough” to achieve acceptable results.

Figure 5 shows how the backlight control signals for  $t_p = 4\text{ms}$  have changed after optimization if compared with basic scanning. It can be seen that the improvement is achieved by turning on more than one LED at the time and by concentrating the light output when this is most convenient to increase  $L_C$ . This can cause the backlight to be turned off for part of the scanning loop, creating a sort of complementary strobing that shuts the backlight off when doing otherwise would be detrimental. The figure also shows that the shape of the square waves controlling the backlight LEDs has not changed and that therefore the optimal result is obtained by properly “shifting” the waves to the most convenient time intervals.

These results are illustrative of the improvements that can be obtained by optimizing backlight scanning, even if some simplifications have been used (i.e. linear transition function). Nevertheless, the model is general enough to allow a more precise simulation of the system.

As a final note, the same experiments have been run for another simulated screen with the same characteristics except for a 10 times lower pixel resolution. The results were nearly identical, showing that it is acceptable to reduce the complexity of the problem by downsampling.

## 5. CONCLUSION

We have proposed a model for backlight scanning which we have used to formulate an optimization problem to minimize crosstalk at a given luminance level. The results show

that better tradeoffs between luminance and crosstalk can be achieved by concentrating the emission of light where this reduces crosstalk. The tradeoff improves with more backlight segments, however the experiments suggest that 4 segments could be sufficient to achieve good tradeoffs. Complexity is not an issue because the value function is linear, the variables are relatively few and there are no particular time requirements to solve the problem, as the optimal scanning signals need to be determined only once. Moreover, downsampling the problem does not affect the final result.

## 6. REFERENCES

- [1] Andrew Woods, “Understanding Crosstalk in Stereoscopic Displays,” in *3DSA (Three-Dimensional Systems and Applications) Conference (Keynote Presentation)*, May 2010.
- [2] Andrew Woods, “How are Crosstalk and Ghosting defined in the Stereoscopic Literature?,” in *Proc. SPIE*, 2011, p. 78630Z.
- [3] Dae-Sik Kim, Young-Ji Ko, Sang-Moo Park, Jong-Hoon Jung, and Sergey Shestak, “Stereoscopic Display Technologies for FHD 3D LCD TV,” in *Proc. SPIE*, 2010, p. 769008.
- [4] Jian-Chiun Liou and Fan-Gang Tseng, “120 Hz low cross-talk stereoscopic display with intelligent LED backlight enabled by multi-dimensional controlling IC,” *Displays*, vol. 30, no. 4-5, pp. 148–154, 2009.
- [5] Haiying Wang, “Correlations between liquid crystal director reorientation and optical response time of a homeotropic cell,” *Journal of Applied Physics*, vol. 95, no. 10, pp. 5502, 2004.
- [6] Pierre Adam, Pascal Bertolino, and Fritz Lebowsky, “Mathematical modeling of the LCD response time,” *Journal of the Society for Information Display*, vol. 15, no. 8, pp. 571–577, 2007.
- [7] Pierre Adam, *Améliorations d’artefacts sur panneaux LCD*. Ph.D. thesis, Institut National Polytechnique de Grenoble, 2008.
- [8] International Committee for Display Metrology Society for Information Display (SID), “Information display measurements standard, version 1.03,” June 2012.

## **A.7 Modeling power-constrained optimal backlight dimming for color displays**

**N. Burini**, E. Nadernejad, J. Korhonen, S. Forchhammer, and X. Wu, “Modeling Power-constrained Optimal Backlight Dimming for Color Displays”, *IEEE/OSA Journal of Display Technology*, 2013. DOI: 10.1109/JDT.2013.2253544

Reference: [10]

## Modeling power-constrained optimal backlight dimming for color displays

Nino Burini, *Student Member, IEEE*, Ehsan Nadernejad, *Student Member, IEEE*, Jari Korhonen, *Member, IEEE*, Soren Forchhammer, *Member, IEEE*, Xiaolin Wu, *Fellow, IEEE*

**Abstract**—We present a framework for modeling color Liquid Crystal Displays (LCDs) having local LED backlight with dimming capability. The proposed framework includes critical aspects like leakage, clipping, light diffusion and human perception of luminance and allows adjustable penalization of power consumption. Based on the framework, we have designed a set of optimization-based backlight dimming algorithms providing a perceptual optimal balance of clipping and leakage, if necessary. The novel algorithms are compared with several other schemes known from the literature, using both objective measures and subjective assessment. The results show that the novel algorithms provide better quality at a given energy level or lower energy at a given quality level.

**Index Terms**—Backlight dimming, Color image, Contrast, High dynamic range (HDR), Image quality, LED backlight, Light leakage, Liquid crystal display (LCD), Optimization, Power saving.

### I. INTRODUCTION

Liquid Crystal Displays (LCDs) are nowadays the most widespread display type, used for several kinds of devices, from digital watches to 3D TV. Liquid Crystals (LCs) are not light emitters but voltage controlled light filters, and therefore require an external light source, typically a built-in *backlight*.

The backlight is usually composed of one or more light sources, a light diffuser distributing the light evenly across the display, and sometimes a light guide, which directs the light before it enters the diffuser. Light sources can be placed behind or at the sides of the diffuser, respectively defining *direct-lit* and *edge-lit* backlights. Today, Light Emitting Diodes (LEDs) are replacing Cold Cathode Fluorescent Lamps (CCFLs) as light sources, thanks to rapidly improving energy efficiency and decreasing price [1]. LED-based backlights can be divided in independently adjustable segments, called *local* backlights, in contrast to *global* backlight, where a single segment covers the whole display; in this work we often use the word LED to refer to backlight segments. Local backlight and LEDs have eased the adoption of *backlight dimming*: with this technique, the backlight can be dimmed to match the image content. When the backlight is divided in local segments, each one can be dimmed to match the brightness of the respective image area. This saves power, since dark areas require less light. Backlight dimming can also reduce *leakage*, a typical problem of LCD: the LCs cannot completely block the light going through them, which makes black pixels look grayish and consequently reduces the contrast ratio. Backlight dimming lowers the amount of light leaking through dark pixels. The possibility to vary the backlight intensity also allows to

increase the number of distinct luminance levels that can be emitted and enable High Dynamic Range (HDR) rendering.

The advantages of backlight dimming come at a cost and create some challenges. Because of the division in segments and the mixing effect of light diffusers, the backlight luminance of each pixel is determined by several LEDs. Because of this, dark and bright pixels are in conflict since the high luminance required by the latter produces leakage in the former, and conversely the low luminance required by the former causes the latter to look dim or in other words to be *clipped*. In particular, the presence of a bright object on dark background can cause a *halo* effect due to leakage. It is therefore not always possible to determine a backlight level that is ideal for all pixels; however, optimization can find the optimal tradeoff given some starting conditions [2]–[4].

Several algorithms have been proposed for backlight dimming. The simplest methods calculate the LED values from global or local image statistics, for example using the maximum, average or square root of the average value of the pixels contained in each segment [5]. These values can be combined together and adjusted with correction tables, as in [6] and [7]. More complex algorithms use features like the global [8] or local histogram [9]. The backlight level in each segment can be improved by considering the neighboring ones [10]. To calculate the backlight more accurately, other algorithms use some level of knowledge of the point spreading function (PSF, also called light spread function [8]), that is how the light spreads from a point source on the diffuser plate [2]–[4].

To obtain optimal results, a backlight dimming algorithm should take into account all the color components and include an accurate model of the display characteristics, like PSFs and (image dependent) leakage, which backlight dimming algorithms often neglect. In this work, we present a model for backlight dimming displays and propose optimization-based algorithms built on top of it, that can calculate screen-specific optimal backlight given some starting parameters. The rest of this paper is organized as follows. Section II explains the model of LCDs with local backlight dimming. In Section III, we present a set of algorithms based on optimization together with others from the literature. Section IV describes the practical experiments we have performed to assess the proposed algorithms and summarizes their results. Finally, Section V provides the concluding remarks.

### II. MODELING LOCAL BACKLIGHT DIMMING IN LCDS

In order to model local backlight dimming, the display characteristics need to be taken into account. The basic

concepts are *transmittance*, the ratio of light that an LC pixel lets through, and *backlight intensity*, the local brightness level of the backlight [3]. These are the input signals to the LCD and determine the resulting image. In a color display, each pixel contains three Red, Green, and Blue (RGB) sub-pixels. The observed backlight intensity at a certain pixel position depends on the physical structure of the diffuser plate and on the light sources. The distribution of light coming from each backlight source is modeled using the PSFs, and the contributions from different light sources are summed to model the total observed backlight at each pixel. Because of leakage, there may be a mismatch between intended and observed transmittance. All these aspects need to be considered in the model.

#### A. Transmittance and leakage

In a locally backlit LCD screen, the observed pixel luminance  $l$  can ideally be expressed as the product of the backlight intensity  $b$  and the LC transmittance  $t$  [3]:

$$l = bt. \quad (1)$$

The values of  $l$ ,  $b$  and  $t$  are normalized to  $[0, 1]$ . If  $b = 0$  there is no light behind the pixel, while if  $b = 1$  the light intensity is maximal. Similarly for the transmittance, if  $t = 1$  the full (normalized) backlight goes through the LC, and if  $t = 0$  the light is fully blocked. However, in practice, leakage prevents LCs to block all the light [3]. Leakage can be modeled linearly by using a parameter  $\varepsilon$ , also called the *leakage factor*, defined as the ratio of light leaking when  $t = 0$  and  $b = 1$ . In presence of leakage, the model of the output luminance  $l$  becomes

$$l = bt + \varepsilon b(1 - t), \quad (2)$$

or alternatively:

$$l = bt_{\varepsilon}, \quad (3)$$

where  $t_{\varepsilon}$  is the observed transmittance, as opposed to ideal transmittance  $t$ ;  $t_{\varepsilon}$  can be expressed as

$$t_{\varepsilon} = (1 - \varepsilon)t + \varepsilon. \quad (4)$$

The model, Eqs. 2-4, allows each pixel to have a different  $\varepsilon$  value, as leakage increases with the viewing angle  $\theta$ . For high accuracy, a vertical and a horizontal viewing angle should be used, as leakage can increase differently in the two directions. The perceived leakage also depends on the ambient light, as it is easier to see in dark environment. For simplicity, in this work we have assumed a constant value of  $\varepsilon$  across the screen.

#### B. Backlight diffusion

Light diffusion in the backlight can be expressed as a function of the intensities of the backlight segments and their PSFs. For each pixel, the backlight  $b$  resulting from a set of LED values is modeled as the sum of all the PSFs multiplied by the corresponding LED intensity, given by

$$b = \sum_{k=1}^N r_k h_k, \quad (5)$$

where  $N$  is the number of segments,  $r_k$  is the intensity of segment  $k$  and  $h_k$  is the value of the PSF of segment  $k$  at the pixel position. In matrix form, Eq. 5 for all pixels is

$$\mathbf{b} = \mathbf{H}\mathbf{r}, \quad (6)$$

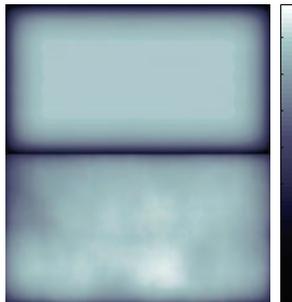


Figure 1. Example of the modeled (top) and the measured backlight (bottom).

where the column vector  $\mathbf{b}$  has a  $b$  value per pixel, the *influence matrix*  $\mathbf{H}$  ( $N$  columns and a row per pixel) represents the PSFs and  $\mathbf{r}$  is a column vector with  $N$  LED values.

In practice, it often may be impractical to define all the individual PSFs. One approximation is to use a high-level segment-based description of the light diffusion, which reduces the amount of data at the cost of precision [4]; another is to use one PSF for all LEDs. The downside of the latter approach is that the segments close to the edges of the screen will trim the PSF and make the modeling of light diffusion inaccurate in that area. Light distribution is typically not perfectly uniform (see Fig. 1). Because of the non-uniformity, not all the pixels can reach the maximum luminance. One solution can be to reduce the *target peak luminance* to some value smaller than the *maximum peak luminance* achievable by the screen. While an excessive reduction of peak luminance is not advisable, lower target peak luminance can improve the backlight uniformity and deliver a more pleasant viewing experience especially when the maximum peak luminance is very high. For this work, we have set the target peak luminance to 99% of the maximum peak luminance, to cancel out peak luminance fluctuations in the central part of the display.

#### C. Backlight-pixel interaction and brightness compensation

The reduction of luminance caused by dimming the backlight can be compensated by increasing the pixel transmittance. This is called *brightness compensation* [11]. In an ideal case, the compensated transmittance  $t_C$  can be solved from Eq. 1 by replacing  $l$  with the target pixel luminance  $l_g$ :

$$t_C = \frac{l_g}{b}. \quad (7)$$

The values of  $t_C$  are limited to lie between 0 and 1, as LCs can only attenuate light. Moreover,  $t_C$  is lower-bounded by  $\varepsilon$

due to leakage. Equation 7 should thus be rewritten as

$$t_C = \begin{cases} 1 & \text{if } l_y/b > 1 \text{ (Cond. I)} \\ 0 & \text{if } l_y/b < \varepsilon \text{ (Cond. II)} \\ (l_y/b - \varepsilon)/(1 - \varepsilon) & \text{otherwise} \end{cases} \quad (8)$$

Condition I defines the occurrence of *clipping*, when the LCs cannot fully compensate the backlight reduction creating *clipped* pixels that appear dimmer than intended. Condition II defines the occurrence of *leakage*, when compensated pixels are brighter than intended. The observed physical transmittance can be computed from Eq. 4, by replacing  $t$  with  $t_C$ .

It should be noted that Eq. 8 describes hard clipping, where pixels are compensated as much as possible. This minimizes the error from the target luminance but on the other hand can visually cause an undesirable posterization effect on bright colors. This problem can be alleviated by using soft clipping, where the clipping curve is smoothed to obtain some differentiation in the posterized areas [11]. Equations 1-8 apply for white backlight and a monochrome LC signal. When more color components are used, as in RGB LCD, the equations have to be applied to all of them. However, the color can change if this is done independently for all channels, as the original ratio between the components may change. A possible solution is to compensate so that the ratio between R, G, and B is maintained; this helps preserving color, at the cost of lower luminance. For simplicity, in this study we have always used hard clipping on the three color channels independently.

#### D. Perception of brightness

The impact of leakage on subjective quality is larger than indicated by the physical luminance error, since the Human Visual System (HVS) perceives it non-linearly: the sensitivity to luminance variations decreases as the luminance increases. In general, a perceptual response can be expressed as

$$l_U = g(l), \quad (9)$$

where  $g$  defines the perceived response of the HVS to the physical luminance  $l$ . The function  $l_U$  is said to be perceptually uniform [12], i.e. a unit step indicates similar perceived difference. Conversely, equal steps of  $l$  correspond to variable step sizes of  $l_U$ . We say that  $l_U$  represents luminance in perceptual domain and  $l$  the luminance in physical domain. It should be noted that the model described in this Section is specified in the physical domain, i.e. physical luminance,  $l$ , is assumed in Eqs. 1-8. In this work, we shall approximate  $g$  with the Gamma function, i.e.

$$l_U = l^{\frac{1}{\gamma}}. \quad (10)$$

The inverse response is simply given by:

$$l = l_U^\gamma. \quad (11)$$

In this work, for simplicity, we approximate the HVS response to luminance with Eq. 10 and  $\gamma = 2.2$ , which coincides with the gamma of the sRGB color space [13] and thereby allows to compare the simulated perceptual output directly with the input image. This is an acceptable choice

when the peak luminance of the display is set to about 100 cd/m<sup>2</sup> [12]; for higher values, larger  $\gamma$  can be used, e.g. 2.2-3.0 are reported to have excellent perceptual performance [14]. In the case of HDR LCDs operating at high peak luminance (1000 cd/m<sup>2</sup> or more), other response functions are more appropriate [12], [15].

### III. OPTIMIZATION-BASED DIMMING ALGORITHMS

This section presents a set of backlight dimming algorithms based on optimization. These algorithms exploit the modeling of backlight and LC transmittance to find the best trade-off between leakage and clipping with the option to constrain power. The backlight can be optimized for one or more color components; a proposal to reduce complexity of the latter case is presented, as well as an approach based on gradient descent.

#### A. Optimal Backlight Dimming

Backlight dimming can be formulated as an image dependent optimization problem. In [2] the target is a clipping-free result; in [4] it is to minimize power limiting clipping below a threshold; in [3] it is the optimal tradeoff between leakage and clipping in the physical domain. In this work, we extend the algorithm in [3] to include power penalization and error weighting. In the algorithms in [2] and [4] leakage is neglected when calculating the backlight. In [4] additionally a strong assumption that the backlight is constant inside segments is made. We propose methods that, besides considering leakage and allowing the use of detailed PSF to determine the backlight, allow to determine the optimal backlight using one of several cost functions thus providing broader generality.

In general, given a multivariate cost function  $f$ , the goal of optimization is finding a combination of variable values minimizing it. For backlight dimming the cost primarily depends on the target output  $y$  and on the actual output  $x$  rendered on the locally backlit LCD. The elements of  $x$  are given by Eq. 3; the backlight  $b$  is given by Eq. 6, the transmittances  $t_k$  are limited between  $\varepsilon$  and 1 (see II-A) and the LED values  $r_k$  between 0 and 1. The initial formulation [3], where  $f$  is the 1-norm ( $l_1$ ) or 2-norm ( $l_2$ ) of  $x - y$ , has been extended to include an error weighting vector  $w$ , with an element per pixel, and a power consumption penalty term given by the product of  $p$  and an adjustable weight  $q$  [16];  $p$  estimates power consumption as the average of the LED values:

$$p = \frac{\sum_{k=1}^N r_k}{N}. \quad (12)$$

This is a good approximation as the LEDs are controlled with PWM signals whose duty cycle is proportional to the emitted luminance. In this work, we have used  $w$  to improve the perceptual result (see Section III-E).

This extended formulation includes [2] and [3] as special cases. We want to minimize a cost function  $f(y, x, w, q, p)$  subject to the model and to constraints on  $t_k$  and  $r_k$ ; the output of the algorithms, presented in this Section, are the optimal  $r_k$  values minimizing  $f$ . Several cost functions will be considered; if  $f$  is linear then the problem belongs to the class of linear problems, i.e. minimization of  $y - x$  in  $l_1$ .

The following sections present three formulations of backlight optimization problems where the cost includes the norm of  $\mathbf{y} - \mathbf{x}$  and power consumption.

Once a linear, or more generally convex, optimization problem has been formulated, it is possible to find a solution using a software solver. For the results reported in this work, we have used the CVX package [17], [18] in Matlab to implement the algorithms of Sections III-B-D.

### B. Single component

The following formulation considers the case where  $\mathbf{y}$  and  $\mathbf{x}$  have one color component, i.e. grayscale images or the luma component of a color frame:

$$\begin{aligned} \min \quad & \|(\mathbf{y} - \mathbf{x}) \circ \mathbf{w}\| + q \times p \\ \text{s.t.} \quad & \mathbf{x} = \mathbf{b} \circ \mathbf{t} \\ & \mathbf{b} = \mathbf{H}\mathbf{r} \\ & \varepsilon \leq t_k \leq 1, \quad k = 1, \dots, mn \\ & 0 \leq r_k \leq 1, \quad k = 1, \dots, N \end{aligned} \quad (13)$$

where the  $\circ$  operator defines element-wise multiplication, and the first two conditions derive from Eqs. 3 and 6, respectively. After optimization, the  $r_k$  values determine the optimal backlight. The problem may be formulated in linear form [3]:

$$\begin{aligned} \min \quad & \|\lambda\| + q \times p \\ \text{s.t.} \quad & \mathbf{b} = \mathbf{H}\mathbf{r} \\ & \lambda \geq ((\varepsilon \circ \mathbf{b}) - \mathbf{y}) \circ \mathbf{w} \\ & \lambda \geq (\mathbf{y} - \mathbf{b}) \circ \mathbf{w} \\ & \lambda \geq 0 \\ & 0 \leq r_k \leq 1, \quad k = 1, \dots, N \end{aligned} \quad (14)$$

If  $q$  is set to 0 and the elements of  $\mathbf{w}$  to 1, the problem is equivalent to [3]. If additionally the elements of  $\varepsilon$  are set to 0, leakage is ignored and the solution will be clipper-free. Assuming small values of  $q$  going to 0, will lead to a minimum energy solution under the clipper-free requirement as in [2].

The error between  $\mathbf{x}$  and  $\mathbf{y}$  is caused by *distorted* pixels, i.e. leaking and clipping pixels. Another source of distortion is the quantization of the LC control signals, usually limited to 8-10 bits. However, leakage and clipping are more critical and the quantization error is ignored in the optimization step.

### C. Color components

The optimization problem can also be formulated for color images and displays, at the cost of higher complexity due to the larger number of variables and constraints. The minimization affects the three RGB color components. However, the monochrome backlight is the same for all components. A specific weight vector may be assigned to each component.

$$\begin{aligned} \min \quad & \left\| \sum_{i=R,G,B} (\mathbf{y}_i - \mathbf{x}_i) \circ \mathbf{w}_i \right\| + q \times p \\ \text{s.t.} \quad & \mathbf{x}_i = \mathbf{b} \circ \mathbf{t}_i; \quad i = R, G, B \\ & \mathbf{b} = \mathbf{H}\mathbf{r} \\ & \varepsilon \leq t_{ki} \leq 1; \quad k = 1, \dots, mn; i = R, G, B \\ & 0 \leq r_k \leq 1; \quad k = 1, \dots, N \end{aligned} \quad (15)$$

### D. Using Min and Max of the color components

This section presents an approach that approximates the full color problem described in the previous section but with the same complexity as the single component formulation. In RGB images, the color of each pixel is given by a triplet of values. The linear formulation of the grayscale optimization problem (Eq. 13) specifies an upper and a lower threshold for the variables  $\lambda$  to be minimized. These thresholds determine leakage and clipping errors. In an RGB triplet, the minimum value corresponds to the leakage threshold, while the maximum value to the clipping threshold. Equation 14 is thus modified by replacing  $\mathbf{y}$  with  $\mathbf{y}_{min}$  or  $\mathbf{y}_{max}$ :  $\mathbf{y}_{min}$  is the vector of the minimum values of each RGB triplet, while  $\mathbf{y}_{max}$  is the vector of the maximum values:

$$\begin{aligned} \min \quad & \|\lambda\| + q \times p \\ \text{s.t.} \quad & \mathbf{b} = \mathbf{H}\mathbf{r} \\ & \lambda \geq ((\varepsilon \circ \mathbf{b}) - \mathbf{y}_{min}) \circ \mathbf{w}_{min} \\ & \lambda \geq (\mathbf{y}_{max} - \mathbf{b}) \circ \mathbf{w}_{max} \\ & \lambda \geq 0 \\ & 0 \leq r_k \leq 1, \quad k = 1, \dots, N \end{aligned} \quad (16)$$

The idea behind this approach is that if the maximum and minimum pixel values can be properly compensated, then the median value can be too. The number of variables  $\lambda$  is reduced to one third, from one per sub-pixel to one per pixel. For a Full HD image, this means going from about 6 million variables to about 2 million. The number of constraints is also reduced by a third. The results may be sub-optimal, but our experiments show that the error increases only slightly (see Section IV-C).

### E. Perception-based error weighting

The optimization problems that have been presented assume  $\mathbf{y}$  and  $\mathbf{x}$  to be linear. However, the HVS does not perceive luminance linearly. This means that the optimal ‘physical’ solution may differ from the optimal perceptual solution. It is possible to mitigate the difference using the weight vector  $\mathbf{w}$  introduced in the extended formulation of Eqs. 13-16.

As mentioned in Section II-D, in this work we approximate the HVS response to luminance using Eq. 10 with  $\gamma = 2.2$ . This function shows that the sensitivity to luminance variation decreases as the luminance increases. We model this sensitivity with the slope of the curve, which in this case gives the weights

$$w = \frac{1}{\gamma} \times \mathbf{y}^{1-\frac{1}{\gamma}}. \quad (17)$$

If, for each pixel, the weight  $w$  is calculated from the target luminance  $y$ , the impact of the error  $\mathbf{y} - \mathbf{x}$  is adjusted accordingly. This increases the influence of leakage errors over clipping errors, reflecting human perception.

### F. Gradient Descent

Given a cost function, the optimal backlight for an image  $\mathbf{y}$  can also be determined through iterative search-based approaches, such as gradient descent. The solution space has as many dimensions as the number of backlight segments. For a given solution, it is possible to calculate the gradient or to estimate it numerically, if a closed form is not available. The

next solution is then obtained by taking a step in the direction opposite to the gradient, since the target is minimization. This process can be iterated as needed, or until an ending condition is met. The solution at iteration  $i + 1$  is calculated by

$$\mathbf{r}_{i+1} = \mathbf{r}_i - s \nabla f(\mathbf{r}_i), \quad (18)$$

where  $\mathbf{r}_i$  is the solution for iteration  $i$ ,  $f$  is the cost function for  $\mathbf{y}$ ,  $\nabla$  is the gradient operator and  $s$  the step size. The minus sign is because the aim is to minimize the cost. The length of  $s$  can be chosen adaptively. In this work, we have used golden section search to find the error-minimizing step for each iteration [19]. Given the gradient of the current solution and Eq. 18, the cost of the next solution ( $\mathbf{r}_{i+1}$ ) can be expressed as a function of  $s$ . Golden section search is a bisection method that allows to find a bracketed minimum of this function iteratively and with a given precision, ensuring fast convergence and reducing the number of function evaluations. The initial bracketing is given by  $s = 0$  and the largest value of  $s$  ensuring that  $\mathbf{r}_{i+1}$  lies within the solution space.

This iterative strategy is very flexible, as it allows to improve existing solutions. Consider for example a video sequence with similar consecutive frames that probably have similar optimal backlight; gradient descent can calculate the solution for the current frame taking the backlight of the previous one as starting solution and adjusting the iterations to the available resources. This would also reduce backlight flickering.

#### G. Reducing complexity

The complexity of these optimization problems can be very high. For example, with Full HD input, Eqs. 14 and 16 have more than two million variables; for Eq. 15 the numbers triple. Thus, it may be necessary to find ways to contain complexity. It has been suggested to exploit the fact that in small areas the backlight is smooth and nearly constant, which would allow to use just one variable per area [3]. A comparable approach has been used in [4], where high precision PSFs are replaced with segment-based high-level approximations of the light diffusion; such approximations might however be too coarse in the case of large backlight segments, like in edge-lit displays. Another solution is to calculate the backlight on a sub-sampled version of the input image. The sub-sampling can be done simply by downscaling the input image, or by adopting more complex strategies that select samples adaptively so that they represent the local characteristic of the image, such as colors, edges and the risk of leakage or clipping; in a previous work we have shown that it is possible to compute nearly-optimal results by using about 10-25% of the input pixels [20]. For the gradient descent approach, an additional parameter is the number of iterations, which can be set as needed; experiments show that the largest improvements happen in the first iterations, therefore good results can be achieved with few loops. For video sequences, using the result from the previous frame as initial guess can provide a fast solution.

#### IV. EXPERIMENTAL RESULTS

We have conducted experiments to evaluate the effectiveness of the proposed approaches. A first set of experiments measured the performance of the reduced complexity algorithm

described in Section III-D. A second set of experiments compared a combination of the proposed approaches against a selection of algorithms from the literature. All the algorithms have been implemented in Matlab. The ones described in Sections III-B-D use the CVX package [17], [18].

The experiments featured the modeling of two locally backlight screens. One is a 55 inch panel with edge-lit backlight having 16 segments placed in 8 rows and 2 columns. The other is modeled on a 47 inch screen from SIM2 [21] with direct-lit backlight having 2202 segments placed in a hexagonal grid. Both screens have full HD resolution (1920x1080 pixels). Our focus has been mainly on the first display.

Power consumption of the backlight is estimated with Eq. 12. The metrics used have been Mean Squared Error (MSE), calculated on the RGB differences between the original and the distorted pixels, and LabPSNR, presented below. The image and backlight values used in the measurements are normalized between 0 and 1. Before being put into the model, the normalized RGB values have been linearized with  $\gamma = 2.2$ , which closely approximates the gamma of the sRGB color space [13]. For each pixel, the target luminance (input for the backlight dimming algorithms) is set to the minimum of the two values given by the linearized input image data and the maximum backlight luminance achievable for the pixel.

#### A. LabPSNR

Since the HVS has different sensitivity for different color components (red, green and blue), the perceived impact of color distortion is difficult to assess with traditional quality measures, such as MSE, weighting different color components in a non-optimal manner. Therefore we propose a metric based on Peak Signal to Noise Ratio (PSNR), but operating in the CIE 1976 L\*a\*b\* color space (denoted Lab for brevity) instead of conventional RGB or YUV spaces. The conversion from linearized (Gamma corrected) sRGB to Lab is defined via the intermediate XYZ color space as follows [13], [22]:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.412 & 0.358 & 0.180 \\ 0.213 & 0.715 & 0.072 \\ 0.019 & 0.119 & 0.950 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (19)$$

$$\begin{aligned} L^* &= 116 \cdot f(Y/Y_n) - 16 \\ a^* &= 500 \cdot [f(X/X_n) - f(Y/Y_n)] \\ b^* &= 200 \cdot [f(Y/Y_n) - f(Z/Z_n)] \end{aligned} \quad (20)$$

where  $f(k) = \begin{cases} k^{1/3} & \text{if } k > 0.008856 \\ 7.787 \cdot k + \frac{16}{116} & \text{otherwise} \end{cases}$

where  $X_n$ ,  $Y_n$  and  $Z_n$  are the X, Y and Z values for the reference white, respectively. The Lab color space has been designed so that any transition of fixed magnitude in the color space approximates an equivalent perceived change, regardless of the direction. Therefore, we can define the color difference  $\Delta E$ , that is the perceived difference between two colors, considering both luminance and chrominance differences [22]:

$$\Delta E = \sqrt{\Delta L^*{}^2 + \Delta a^*{}^2 + \Delta b^*{}^2}, \quad (21)$$

where  $\Delta L^*$ ,  $\Delta a^*$  and  $\Delta b^*$  define the differences between the original and distorted pixel measured for  $L^*$ ,  $a^*$  and

$b^*$  components. Using PSNR, we can define *LabPSNR* by replacing conventional MSE by mean squared  $\Delta E$ :

$$LabPSNR = 10 \cdot \log_{10} \left( \frac{(\Delta E_{max})^2}{\frac{1}{mn} \sum_{i=1}^n \sum_{j=1}^m \Delta E(i, j)^2} \right), \quad (22)$$

where  $\Delta E(i, j)$  is the color difference at pixel position  $(i, j)$ , given by Eq. 21,  $m$  and  $n$  define the image dimensions, and  $\Delta E_{max}$  is the difference between black and white, i.e. normalized sRGB triplets (0,0,0) and (1,1,1); its value is 100.

### B. Backlight dimming algorithms

We have selected some backlight dimming algorithms presented in the literature to compare our approach against other algorithms, besides the conventional full backlight and the simplest algorithms described in the introduction [5]. The algorithms introduced by Cho et al. [6] and Zhang et al. [7] first calculate the backlight intensity combining average and maximum luminance of the input image, then add a correction term; in [6] it is based on the difference between maximum and minimum values of each backlight segment; in [7] it is based on the estimated loss of detail occurring when the backlight is reduced to the average luminance of the input image. In [8], Lin et al. propose to use the inverse cumulative distribution of the global histogram to map, for each segment, a weighted mean of the average and maximum pixel value to backlight intensity. Kang et al. [9] presented an algorithm where multi-histograms are used to analyze the pixel distribution for the RGB components of the input image and set the backlight to limit the occurrence of clipping. The algorithm proposed by Kim et al. [10] calculates the backlight segment intensity by analyzing the neighboring segments and by comparing leakage and clipping measures. Albrecht et al. [2] introduced a clipper-free algorithm which minimizes the power consumption under this constraint. This formulation may be seen as a limiting case of Eq. 15 by setting  $\varepsilon$  and  $q$  to 0. In the first step of the algorithm, lower bounds are set for each backlight segment, depending on the image content and the PSF. The second (optional) step is iterative and finds, in each loop, the most unsatisfied pixel (the pixel that requires the largest increase in luminance to avoid clipping) and increases the intensity of the most influential LED to satisfy it. The final step scans the pixels of each segment in a specific order determined by the PSF and adjusts the LEDs to make sure no pixel is clipped. In our implementation we have included the first two steps, which produce a clipper-free result.

### C. Performance of the Min-Max color approach

This experiment aimed at assessing the performance of the reduced-complexity color optimization approach based on the maximum and minimum RGB values (Eq. 16) compared with the full RGB problem (Eq. 15). The optimal backlight has been calculated on both modeled screens, at different downscaling factors (4, 5, 6 and 8 for the edge-lit panel, 8 and 10 for the direct-lit panel), for two leakage factors  $\varepsilon$  valued 0.0002 and 0.001 and for both  $\ell_1$  and  $\ell_2$ . The power penalty  $q$  was

Table 1  
COMPARISON OF FULL AND REDUCED COMPLEXITY COLOR  $\ell_2$  MINIMIZATION ON EDGE-LIT (TOP) AND DIRECT-LIT (BOTTOM) PANELS.

	$\varepsilon = 0.0002$		$\varepsilon = 0.001$	
	Power	MSE	Power	MSE
Full color	74.47	2.731 E-4	37.28	9.202 E-4
Color Min-Max	76.87	2.820 E-4	39.84	9.574 E-4
Variation	2.40	3.26%	2.56	4.04%

	$\varepsilon = 0.0002$		$\varepsilon = 0.001$	
	Power	MSE	Power	MSE
Full color	38.55	4.584 E-5	31.78	1.247 E-4
Color Min-Max	39.30	4.807 E-5	32.46	1.345 E-4
Variation	0.75	4.86%	0.68	7.84%

set to zero. The input images have been downsampled with bicubic interpolation. The input data set included seven images (Man, Stars, City1 and City2 shown in Fig. 2, one of night fireworks, one frame from the ending titles of a movie and a frame from a high contrast animation sequence) which, in past experiments, proved to be challenging by showing a higher error after optimization compared to other images. Table 1 shows the results for downscaling factor 4 (edge-lit screen) and 10 (direct-lit screen). In  $\ell_2$  minimization, MSE increases at most by 7.84%. Power consumption, normalized and multiplied by 100, increases as well with values between 0.68 and 2.56. Results for other downscaling factors and for  $\ell_1$  minimization are analogous and not reported here. Considering the significant reduction of variables, these results support our statement that the reduced-complexity approach can be a viable alternative to the full problem. As would be expected, a higher leakage factor yields lower power consumption as the leakage can only be reduced by dimming the backlight.

### D. Optimal backlight and comparison with other algorithms

In the second set of experiments, we have calculated the backlights minimizing  $\ell_1$  and  $\ell_2$  and compared against the algorithms presented in Section IV-B on a set of 32 images: the 24 images of the Kodak True Color Image Suite [23] and 8 more shown in Fig. 2. We have used gradient descent (Eq. 18) to calculate the perceptual optima at Full HD resolution; the cost function is as in Eq. 15 but with  $y_j$  and  $x_i$  in the perceptual domain (approximated with Eq. 10) and with all the elements of  $w_i$  set to 1. The starting solution was generated with the Min-Max approach (Eq. 16) minimizing the same norm and with perception-based error weighting enabled at a downscaling factor of 5. It should be noted that the cost functions are not identical as the gradient descent approach optimizes directly in the perceptual domain, while the Min-Max approach uses the error-weighting vector  $w$ . Figure 3 shows a plot comparing the MSE of the proposed approach with  $\ell_2$  minimization with that of other algorithms. The optimization has been performed with different power penalty values  $q$  that provided solutions at power levels close to those of the other algorithms to enable fair comparisons. The results are averaged over the test set of 32 images and, for the proposed approach, at the same  $q$ . Figure 4 shows the same comparison in LabPSNR. The LED and LC values have also been quantized as this is necessary to display them on real

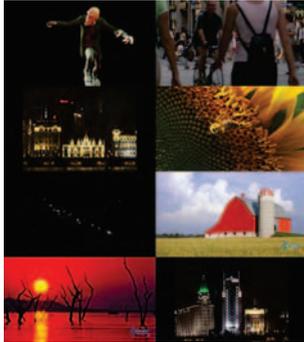


Figure 2. Sample of images used for the experiments: (left to right, top to bottom) Man, Pedestrian, City1, Sunflower, Stars, Barn, Sunset and City2.

screens. Quantizing LED values does not have a significant impact but quantizing the LC does; in Fig. 4 we show results for LC values quantized to 8 and 10 bits. Quantization does not have a large effect on overall MSE and is omitted in Fig. 3. The average algorithm has been removed from Fig. 3 due to its high error (1.99 E-3). The figures show that at a given power level the proposed approach achieves the best result.

Table II reports the MSE values from Fig. 3 split into leakage and clipping contributions, with extra data for  $\epsilon = 0.0002$ . Clipping values are for  $\epsilon = 0.001$ . The  $\epsilon$  value affects the LED values output by the proposed approach but not the other algorithms, for which clipping is the same for both  $\epsilon$ . Clipping for the proposed approach is  $2.68 \text{ E-}6$  (13.01%) at  $q = 0$  and  $3.60 \text{ E-}5$  (74.01%) at  $q = 100$  if optimizing for  $\epsilon = 0.0002$ . The dotted line in Fig. 3 depicts the clipping contribution for each  $q$  of our optimization based algorithm. The table confirms that leakage has more impact for higher  $\epsilon$ .

An additional result is presented in Fig. 5: optimization in  $\ell_2$  was performed for direct-backlit screen and only on the images shown in Fig. 2. Since calculating the gradient is very time-demanding due to the high number of segments, we only run the initial optimization with CVX [17], [18] at a downscaling factor of 5. The average algorithm is again omitted due to the excessive MSE (3.31 E-3). Also on this display, the proposed approach achieves the lowest error at all power levels.

These results altogether show that high power algorithms tend to render with higher fidelity but neglect leakage, which is the main cause of distortion, while low power algorithms suffer from clipping. The proposed optimization-based approaches reduce distortion in all cases finding the optimal tradeoff at a given power level. This reduces color distortion, as shown by the LabPSNR data in Fig. 4. The same figure shows that the

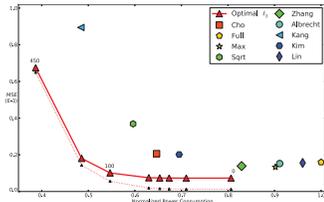


Figure 3. Distortion (MSE) vs. power trade-off of the proposed approach (curve) compared with other algorithms on edge-lit 16 segments backlight. Labels indicate power penalty. The dotted line plots the contribution of clipping to MSE for optimal  $\ell_2$ .

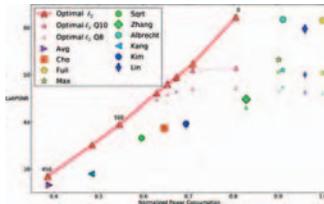


Figure 4. Quality (LabPSNR) vs. power trade-off of the proposed approach (curve) compared with other algorithms on edge-lit 16 segments backlight. Labels indicate power penalty. The smaller markers indicate quantized results for 10 bits (middle size) and 8 bits (smallest); power levels and symbols are the same as the non-quantized result.

quantization error tends to dominate leakage and clipping error at high power levels, as revealed by the tendency of the curves to saturate. Quantization is less important at lower power.

E. Subjective experiments

Many studies have proven that simple distortion measures, such as MSE and PSNR, do not always predict the perceived

Table II  
CONTRIBUTION OF LEAKAGE AND CLIPPING TO MSE (EDGE-LIT).

	$\epsilon = 0.0002$		$\epsilon = 0.001$		Clipping
	Leakage	%	Leakage	%	
Avg	2.70 E-6	0.14	1.32 E-5	0.66	1.97 E-3
Cho	1.66 E-5	11.26	7.54 E-5	36.57	1.31 E-4
Full	3.51 E-5	100	1.60 E-4	100	0
Max	2.91 E-5	98.88	1.33 E-4	99.75	3.31 E-7
Sqrt	7.33 E-6	2.15	3.46 E-5	9.38	3.34 E-4
Zhang	2.88 E-5	77.86	1.31 E-4	94.11	8.18 E-6
Albrecht	3.36 E-5	100	1.52 E-4	100	0
Kim	1.66 E-5	1.99	7.46 E-5	8.25	8.19 E-4
Kim	1.97 E-5	14.93	8.93 E-5	44.26	1.12 E-4
Lin	3.41 E-5	97.38	1.55 E-4	99.41	9.18 E-7
Opt q=0	1.79 E-5	86.99	6.03 E-5	98.48	1.28 E-5
Opt q=100	1.21 E-5	25.09	4.57 E-5	45.09	5.56 E-5

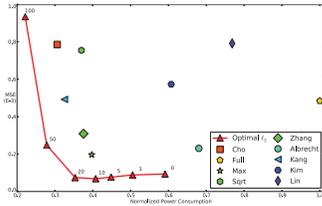


Figure 5. Distortion (MSE) vs. power trade-off of the proposed approach (curve) compared with other algorithms on direct-lit 2202 segments backlight (downscaled 5 times). Labels indicate power penalty.

subjective quality accurately. However, accuracy is highly dependent on the distortion type and, in our case, also on the accuracy of the backlight model, in particular leakage and PSFs. In order to assess the visual quality and to verify that the subjectively perceived image quality follows the measured distortion in the model, we have performed a subjective test to compare our gradient descent approach with other algorithms; the backlight of the edge-lit display is simulated and shown on the direct-lit SIM2 display. The borders (ca. 30 pixels on each side) of the screen where covered to hide backlight non-uniformities close to the edges.

Since subjective scoring schemes are not well suited for distinguishing small visual differences between images, we have adopted a pairwise comparison method, where the subjects choose the preferred image out of two versions of it, produced with a different backlight dimming scheme. In order to limit the duration of the experiment, but yet to have different image types represented, we have used a subset of seven images, where three images have been chosen among those used in the objective evaluation (Stars image in Fig. 2, Beach and Parrot from the Kodak dataset), and the other four are frames from the Volcano and Diver video sequences from [24], and the images Exotic and Lizard from [25]. The latter four were included as they have saturated colors and a high contrast.

Sixteen test subjects performed the test, all of them naive regarding the purpose of the test and not experts in backlight dimming. The participants were 12 men and 4 women, of age ranging from 22 to 30. During the test session, each observer was allowed to freely switch between the two compared images before making the binary choice of preference. The reader may refer to [26] for more details of the test arrangement.

The results from pairwise comparisons were transformed into a subjective rank order, and the correlation between the rank order based on objective results (average LabPSNR) and the subjective preference was studied. The results are summarized in Table III. Except for one outlier case (Stars), the results show a fairly good match between subjective and objective results in terms of Spearman rank order correlation coefficient (SROCC). The average SROCC is 0.80. Gradient descent with low power weight ( $q=0$  or  $q=1$ ) outperforms all

Table III  
PERFORMANCE COMPARISON OF BACKLIGHT DIMMING ALGORITHMS. ALGORITHMS ARE LISTED IN SUBJECTIVE PREFERENCE ORDER FOR EACH IMAGE;  $GDqX$  REFERS TO THE GRADIENT DESCENT ALGORITHM WITH POWER WEIGHT  $q=X$ ; CENTRAL COLUMNS ARE LABPSNR, RIGHT COLUMNS ARE NORMALIZED POWER CONSUMPTION.

Exotic (SROCC 0.96)		Lizard (SROCC 0.89)			
$GDq0$	45.17	0.948	$GDq1$	46.46	0.870
Albrecht	45.78	0.985	Albrecht	47.35	0.981
Zhang	38.01	0.932	Zhang	36.28	0.904
$GDq200$	33.52	0.763	$GDq400$	37.06	0.644
Kang	29.32	0.752	Cho	25.35	0.684
$GDq400$	23.86	0.578	Kang	24.57	0.658
Cho	22.76	0.655			
Parrot (SROCC 0.96)		Beach (SROCC 0.89)			
$GDq0$	51.42	0.813	$GDq1$	51.91	0.730
Albrecht	51.74	0.840	Albrecht	52.92	0.855
Zhang	50.00	0.811	Cho	39.11	0.767
$GDq100$	44.92	0.648	Zhang	45.25	0.800
Cho	38.50	0.660	$GDq800$	23.17	0.483
$GDq600$	25.37	0.440	Kang	20.42	0.448
Kang	25.02	0.447			
Volcano (SROCC 0.83)		Diver (SROCC 0.70)			
Albrecht	38.98	0.660	$GDq0$	46.90	0.702
$GDq0$	38.19	0.513	$GDq400$	30.12	0.546
Zhang	38.04	0.767	Albrecht	44.11	0.825
Cho	25.48	0.439	Zhang	32.54	0.740
Kang	22.62	0.388	Cho	23.66	0.560
$GDq400$	26.39	0.324			
Stars (SROCC 0.40)					
$GDq0$	39.29	0.304			
Albrecht	36.89	0.982			
Kang	38.58	0.504			
Cho	38.34	0.499			



Figure 6. Displayed results of the Stars image (as in Table III) for backlight algorithm Albrecht (left) and gradient descent ( $GDq0$ , right).

the other algorithms in terms of subjective preference in all cases except one (Volcano), where Albrecht is preferred, however the latter requires significantly higher power consumption. Another important observation is that both subjective and objective results show the gradient descent algorithm outperforming all the other algorithms with the same or lower power consumption. Figure 6 shows two pictures of the displayed results of the Albrecht and gradient descent algorithms for the Stars image. For the gradient descent, leakage is less annoying.

## V. CONCLUSION

We have presented a model for LCD with LED local backlight dimming that includes important modeling aspects like leakage, clipping, PSFs, color and human perception of luminance. Based on it, we have designed optimization-based dimming algorithms, considering both leakage and clipping, and compared them to other approaches. The model allows the proposed methods to optimize several cost functions and

to obtain the best trade-offs between quality and power consumption at all power levels. In particular, the approach based on gradient descent is very versatile and powerful, as it allows to optimize for non-linear cost functions at Full HD resolution.

REFERENCES

- [1] M. Anandan, "Progress of LED backlights for LCDs," *J. Soc. Inf. Display*, vol. 16, no. 2, pp. 287-310, 2008.
- [2] M. Abbrecht, A. Kahrenbauer, T. Jung, and C. Xu, "Sorted Sector Covering Combined with Image Condensation: An Efficient Method for Local Dimming of Direct-Lit and Edge-Lit LCDs," *IEICE Trans. Electronics*, vol. E93-C, no. 11, pp. 1556-1563, 2010.
- [3] X. Shu, X. Wu, and S. Forchhammer, "Optimal Local Dimming for LC Image Formation with Controllable Backlighting," *IEEE Trans. Image Process.*, vol. 22, no. 1, Jan 2013.
- [4] J.-J. Hong, S.-E. Kim, and W.-J. Song, "A Clipping Reduction Algorithm Using Backlight Luminance Compensation for Local Dimming Liquid Crystal Displays," *IEEE Trans. Consum. Electron.*, vol. 56, no. 1, pp. 240-246, February 2010.
- [5] H. Seetzen, W. Heidrich, W. Stuerzlinger, G. Ward, L. Whitehead, M. Trantacoste, A. Ghosh, and A. Vorozcovs, "High dynamic range display systems," *ACM Trans. Graphics*, vol. 23, no. 3, p. 760, 2004.
- [6] H. Cho and O.-K. Kwon, "A backlight dimming algorithm for low power and high image quality LCD applications," *IEEE Trans. Consum. Electron.*, vol. 55, no. 2, pp. 839-844, 2009.
- [7] X.-B. Zhang, R. Wang, D. Dong, J.-H. Han, and H.-X. Wu, "Dynamic Backlight Adaptation Based on the Details of Image for Liquid Crystal Displays," *J. Display Technol.*, vol. 8, no. 2, pp. 108-111, Feb 2012.
- [8] F.-C. Lin, Y.-P. Huang, L.-Y. Liao, C.-Y. Liao, H.-P. D. Shieh, T.-M. Wang, and S.-C. Yeh, "Dynamic Backlight Gamma on High Dynamic Range LCD TVs," *J. Display Technol.*, vol. 4, no. 2, pp. 139-146, 2008.
- [9] S.-J. Kang and Y. H. Kim, "Multi-Histogram-Based Backlight Dimming for Low Power Liquid Crystal Displays," *J. Display Technol.*, vol. 7, no. 10, pp. 544-549, Oct 2011.
- [10] S.-E. Kim, J.-Y. An, J.-J. Hong, T. W. Lee, C. G. Kim, and W.-J. Song, "How to reduce light leakage and clipping in local-dimming liquid-crystal displays," *J. Soc. Inf. Display*, vol. 17, no. 12, p. 1051, 2009.
- [11] L. Kerofsky and S. Daly, "Brightness preservation for LCD backlight dimming," *J. Soc. Inf. Display*, vol. 14, no. 12, p. 1111, 2006.
- [12] T. O. Aydin, R. Mantiuk, and H.-P. Seidel, "Extending quality metrics to full luminance range images," in *Proc. SPIE*, vol. 6806, 2008, pp. 6806-10.
- [13] "sRGB Color Space," <http://www.w3.org/Graphics/Color/sRGB/>.
- [14] C. Poynton, *A Technical Introduction to Digital Video*. John Wiley & Sons, 1996, ch. 6, p. 95.
- [15] J. Korhonen, N. Burini, S. Forchhammer, and J. Meldgaard Pedersen, "Modeling LCD displays with local backlight dimming for image quality assessment," in *Proc. SPIE*, vol. 7866, 2011, p. 843607.
- [16] N. Burini, E. Nadernejad, J. Korhonen, S. Forchhammer, and X. Wu, "Image dependent energy-constrained local backlight dimming," in *Proc. IEEE ICIP*, 2012.
- [17] M. Grant and S. Boyd, "CVX: Matlab software for disciplined convex programming, version 1.21," <http://cvx.com/cvx>, Apr 2011.
- [18] M. Grant and S. Boyd, "Graph implementations for nonsmooth convex programs," in *Recent Advances in Learning and Control*, ser. Lecture Notes in Control and Information Sciences, V. Blondel, S. Boyd, and H. Kimura, Eds. Springer-Verlag Limited, 2008, pp. 95-110.
- [19] W. H. Press, S. A. Teukolski, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C, 2nd ed.* Cambridge University Press, 1992.
- [20] N. Burini, E. Nadernejad, J. Korhonen, S. Forchhammer, and X. Wu, "Speechup of optimization-based approach to local backlight dimming of HDR displays," in *Proc. SPIE*, vol. 8436, 2012, pp. 84360B-1.
- [21] "SIM2 HDR TV," <http://www.sim2.com/HDR/>.
- [22] J.-R. Ohm, *Multimedia Communication Technology: Representation, Transmission and Identification of Multimedia Signals*. Springer, 2004.
- [23] "Kodak True Color Image Suite," <http://f0k.us/graphics/kodaki/>.
- [24] "The Consumer Digital Video Library," <http://www.cdvl.org>.
- [25] A. Olmos and F. A. A. Kingdom, "A biologically inspired algorithm for the recovery of shading and reflectance images," *Perception*, vol. 33, pp. 1463-1473, 2004.
- [26] E. Nadernejad, N. Burini, J. Korhonen, S. Forchhammer, and C. Mantel, "Adaptive local backlight dimming algorithm based on local histogram and image characteristics," in *Proc. SPIE*, vol. 8652, 2013, p. 86520V.



**Nino Burini** (M'10) received his B.S. and M.S. degrees in computer science and engineering from the University of Ferrara, Italy, in 2006 and 2009, respectively. He is currently pursuing his Ph.D. degree at DTU Fotonik, the Department of Photonics Engineering of the Technical University of Denmark, Kongens Lyngby. His research interests include LCD backlight, display technology, image and video coding and processing.



**Ehsan Nadernejad** (M'12) received his B.S. and M.S. degrees in electronic engineering from Mazandaran University, Iran, in 2003 and 2007, respectively. He is working toward the Ph.D. degree at DTU Fotonik, Department of Photonics Engineering of the Technical University of Denmark, Kongens Lyngby. His research interests include signal and image processing, video coding, post-processing of image and video signals and display technology.



**Jari Korhonen** (M'05) received his M.S. (Eng.) degree in information engineering from University of Oulu, Finland, in 2001, and Ph.D. degree in telecommunications from Tampere University of Technology, Finland, in 2006. He is currently an Assistant Professor at DTU Fotonik, Technical University of Denmark, since 2010. His research interests cover both telecommunications and signal processing aspects in multimedia communications, including visual quality assessment, error control mechanisms, and multimedia transmission over wireless networks.



**Soren Forchhammer** (M'04) received the M.S. degree in engineering and the Ph.D. degree from the Technical University of Denmark, Lyngby, in 1984 and 1988, respectively. Currently, he is a Professor with DTU Fotonik, Technical University of Denmark, where he has been since 1988. He is Head of the Coding and Visual Communication Group at DTU Fotonik. His interests include source coding, image and video coding, distributed source coding, processing for image displays, two-dimensional information theory, and visual communications.



**Xiaolin Wu** (SM'96-F'10) received the B.S. degree from Wuhan University, China, in 1982 and the Ph.D. degree from the University of Calgary, Canada, in 1988, both in computer science. He started his academic career in 1988, and he has been with the faculty of the University of Western Ontario, Canada, and of New York Polytechnic University, Brooklyn. He is currently Professor at the Department of Electrical and Computer Engineering, McMaster University, Canada, where he holds the Natural Sciences and Engineering Research Council of Canada/DALSA Industrial Research Chair in Digital Cinema. He has published more than 180 research papers and holds two patents related to his fields of interest. His research interests include image processing, multimedia compression, joint source-channel coding, multiple description coding, and network-aware visual communication. Dr. Wu currently serves as the associate editor of the IEEE Transactions on Image Processing.

# Appendix B

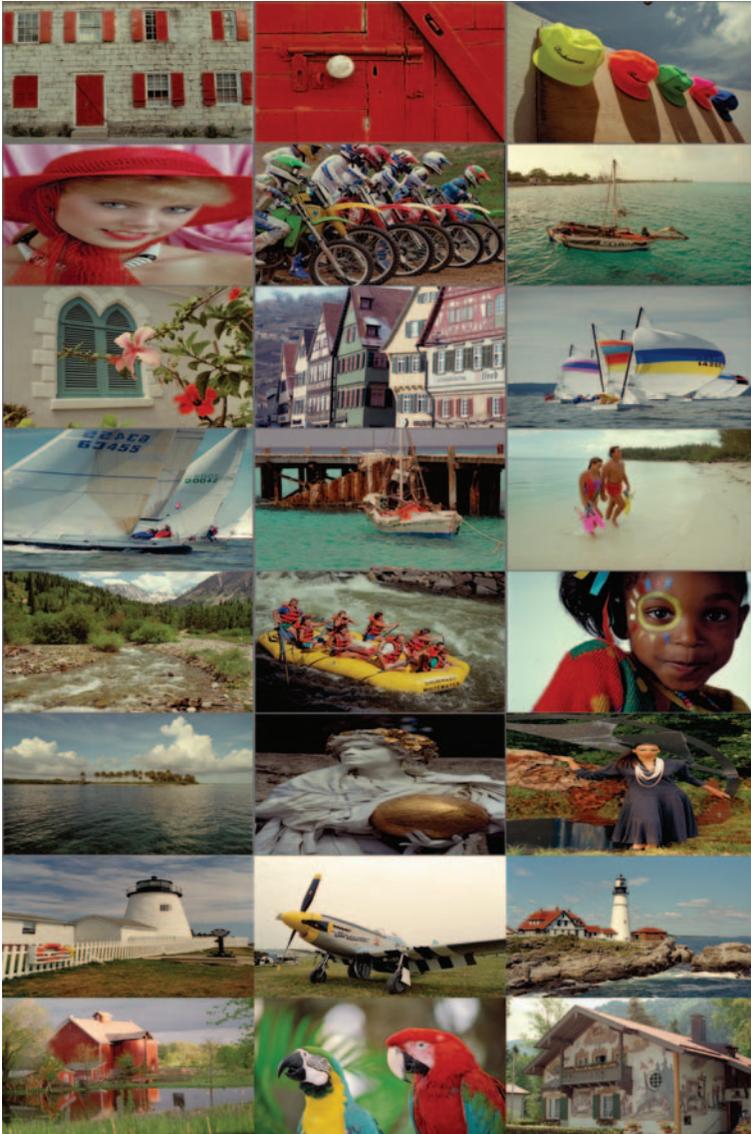
## Testsets



**Figure B.1:** This testset is called “ICIP” because it was originally used for the experiments whose results were submitted to IEEE ICIP 2012; its images are called (left to right, top to bottom) *Man*, *Pedestrian*, *City1*, *Sunflower*, *Stars*, *Barn*, *Sunset* and *City2*.



**Figure B.2:** This testset is called “ICIP” because it was originally used for the experiments whose results were submitted to IEEE ICIP 2012; its images are called (left to right, top to bottom) *Man*, *Pedestrian*, *City1*, *Sunflower*, *Stars*, *Barn*, *Sunset* and *City2*. Grayscale version.



**Figure B.3:** Kodak True Color testset [56]. The images are called  $k01$ ,  $k02$ ,  $k03$ ,  $k04$ ,  $\dots$ ,  $k23$ ,  $k24$ , starting from the top-left then going left to right and top to bottom.



**Figure B.4:** Kodak True Color testset [56]. The images are called  $k01$ ,  $k02$ ,  $k03$ ,  $k04$ ,  $\dots$ ,  $k23$ ,  $k24$ , starting from the top-left then going left to right and top to bottom. Grayscale version.



**Figure B.5:** Four images used for experiments: *Exotic flower*, *Lizard*, *Diver*, *Volcano*. The first two are found in [57], the other two in [58].



**Figure B.6:** First frames of 9 video sequences used for experiments: *Akiyo*, *Bus*, *Coastguard*, *Container*, *Cycling*, *Foreman*, *Hall*, *Mobile*, *Mother*, *News*, *Paris* and *Silent*.

# References

- [1] S. Forchhammer, M. Danieli, **N. Burini**, M. Zamarin, and A. Ukhanova, “Maximizing entropy of image models for 2-D constrained coding”, in *Proceedings of Workshop on Information Theoretic Methods in Science and Engineering (WITMSE)*, 2010.
- [2] J. Korhonen, **N. Burini**, S. Forchhammer, and J. M. Pedersen, “Modeling LCD displays with Local Backlight Dimming for Image Quality Assessment”, in *Proceedings of SPIE*, vol. 7866, 2011, p. 786 607.
- [3] **N. Burini**, E. Nadernejad, J. Korhonen, S. Forchhammer, and X. Wu, “Speedup of Optimization-based Approach to Local Backlight Dimming of HDR Displays”, in *Proceedings of SPIE*, vol. 8436, 2012, 84360B.
- [4] J. Korhonen, **N. Burini**, J. You, and E. Nadernejad, “How to Evaluate Objective Video Quality Metrics Reliably”, in *International Workshop on Quality of Multimedia Experience (QoMEX)*, Jul. 2012, pp. 57–62.
- [5] **N. Burini**, E. Nadernejad, J. Korhonen, S. Forchhammer, and X. Wu, “Image Dependent Energy-Constrained Local Backlight Dimming”, in *IEEE International Conference on Image Processing (ICIP)*, Sep. 2012, pp. 2797–2800.
- [6] E. Nadernejad, **N. Burini**, J. Korhonen, S. Forchhammer, and C. Mantel, “Adaptive Local Backlight Dimming Algorithm Based on Local Histogram and Image Characteristics”, in *Proceedings of SPIE*, vol. 8652, 2013, p. 86520V.

- 
- [7] E. Nadernejad, J. Korhonen, S. Forchhammer, and **N. Burini**, “Enhancing Perceived Quality of Compressed Images and Video with Anisotropic Diffusion and Fuzzy Filtering”, *Signal Processing: Image Communication*, vol. 28, no. 3, pp. 222–240, Mar. 2013.
- [8] E. Nadernejad, **N. Burini**, and S. Forchhammer, “Adaptive Deblocking and Deringing of H.264-AVC Video Sequences”, in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2013, pp. 2508–2512.
- [9] **N. Burini**, X. Shu, L. Jiao, S. Forchhammer, and X. Wu, “Optimal Backlight Scanning for 3D Crosstalk Reduction in LCD TV”, in *IEEE International Conference on Multimedia & Expo (ICME)*, To be presented, 2013.
- [10] **N. Burini**, E. Nadernejad, J. Korhonen, S. Forchhammer, and X. Wu, “Modeling Power-constrained Optimal Backlight Dimming for Color Displays”, *IEEE/OSA Journal of Display Technology*, 2013. DOI: 10.1109/JDT.2013.2253544.
- [11] C. Mantel, **N. Burini**, E. Nadernejad, J. Korhonen, S. Forchhammer, and J. Meldgaard Pedersen, “Controlling power consumption for displays with backlight dimming”, *IEEE/OSA Journal of Display Technology*, 2013. DOI: 10.1109/JDT.2013.2260131.
- [12] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, “Image Quality Assessment: from Error Visibility to Structural Similarity”, *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.
- [13] X. Shu, X. Wu, and S. Forchhammer, “Optimal Local Dimming for LC Image Formation with Controllable Backlighting”, *IEEE Transactions on Image Processing*, vol. 22, no. 1, Jan. 2013.
- [14] C. P. Halsted, “Brightness, Luminance, and Confusion”, *Information Display*, Mar. 1993.
- [15] H. Sugiura, “Multiple Primary Color Backlights”, in *LCD Backlights*, S. Kobayashi, S. Mikoshiba, and S. Lim, Eds. Wiley, 2009, ch. 3, pp. 33–47.
- [16] W. Den Boer, “Introduction”, in *Active Matrix Liquid Crystal Displays*. Elsevier, 2005, ch. 1, pp. 1–22.

- 
- [17] C. Poynton, “A technical introduction to digital video”, in. John Wiley & Sons, 1996, ch. 6, p. 95.
- [18] T. Yamatomo, “Improvement of Moving Picture Quality by Means of Backlight Control”, in *LCD Backlights*, S. Kobayashi, S. Mikoshiba, and S. Lim, Eds. Wiley, 2009, ch. 2, pp. 21–32.
- [19] S. Y. Lee, “Technical Trends and Requirements/Specifications for LCD TV Backlights”, in *LCD Backlights*, S. Kobayashi, S. Mikoshiba, and S. Lim, Eds. Wiley, 2009, ch. 1, pp. 3–20.
- [20] The European Commission, “Commission Regulation (EC) No 642/2009 of 22 July 2009 implementing Directive 2005/32/EC of the European Parliament and of the Council with regard to ecodesign requirements for televisions”, *Official Journal of the European Union*, vol. L191, pp. 42–52, Jul. 23, 2009.
- [21] E. H. A. Langendijk, R. Muijs, and W. van Beek, “Contrast Gain and Power Savings Using Local Dimming Backlights”, *Journal of the Society for Information Display*, vol. 16, pp. 1237–1242, 2008.
- [22] “Progress of LED backlights for LCDs”, *Journal of the Society for Information Display*, vol. 16, no. 2, pp. 287–310, 2008.
- [23] J. T., “Ultra-Slim LCD TV Enabled by an Edge-Lit LED Backlight Systems”, *Information Display*, vol. 24, no. 11, pp. 14–17, 2008.
- [24] T. Funamoto, T. Kobayashi, and T. Murao, “High-Picture-Quality Technique for LCD televisions: LCD-AI”, in *Proceedings of the International Display Workshop*, 2001, pp. 1157–1158.
- [25] H. Seetzen, W. Heidrich, W. Stuerzlinger, G. Ward, L. Whitehead, M. Trantacoste, A. Ghosh, and A. Vorozcovs, “High Dynamic Range Display Systems”, *ACM Transaction on Graphics*, vol. 23, no. 3, p. 760, 2004.
- [26] H. Cho and O.-K. Kwon, “A Backlight Dimming Algorithm for Low Power and High Image Quality LCD Applications”, *IEEE Transactions on Consumer Electronics*, vol. 55, no. 2, pp. 839–844, 2009.

- [27] X.-B. Zhang, R. Wang, D. Dong, J.-H. Han, and H.-X. Wu, “Dynamic Backlight Adaptation Based on the Details of Image for Liquid Crystal Displays”, *IEEE/OSA Journal of Display Technology*, vol. 8, no. 2, pp. 108–111, Feb. 2012.
- [28] H. Nam, “Low Power Active Dimming Liquid Crystal Display with High Resolution Backlight”, *Electronics Letters*, vol. 47, p. 538, 9 2011.
- [29] S.-E. Kim, J.-Y. An, J.-J. Hong, T. W. Lee, C. G. Kim, and W.-J. Song, “How to Reduce Light Leakage and Clipping in Local-dimming Liquid-crystal Displays”, *Journal of the Society of Information Display*, vol. 17, no. 12, p. 1051, 2009.
- [30] S.-J. Kang and Y. H. Kim, “Image Integrity-based Gray-Level Error Control for Low Power Liquid Crystal Displays”, *IEEE Transactions on Consumer Electronics*, vol. 55, no. 4, pp. 2401–2406, 2009.
- [31] ———, “Multi-Histogram-Based Backlight Dimming for Low Power Liquid Crystal Displays”, *IEEE/OSA Journal of Display Technology*, vol. 7, no. 10, pp. 544–549, Oct. 2011.
- [32] H. Chen, J. Sung, T. Ha, and Y. Park, “Locally Pixel-compensated Backlight Dimming on LED-backlit LCD TV”, *Journal of the Society of Information Display*, vol. 15, no. 12, pp. 981–988, 2007.
- [33] C.-C. Lai and C.-C. Tsai, “Backlight Power Reduction and Image Contrast Enhancement Using Adaptive Dimming for Global Backlight Applications”, *IEEE Transactions on Consumer Electronics*, vol. 54, no. 2, pp. 669–674, 2008.
- [34] H. Cho, B. Chul Cho, H. Jung Hong, E.-Y. Oh, and O.-K. Kwon, “A Color Local Dimming Algorithm for Liquid Crystals Displays Using Color Light Emitting Diode Backlight Systems”, *Optics & Laser Technology*, vol. 47, pp. 80–87, 2013.
- [35] F.-C. Lin, Y.-P. Huang, L.-Y. Liao, C.-Y. Liao, H.-P. D. Shieh, T.-M. Wang, and S.-C. Yeh, “Dynamic Backlight Gamma on High Dynamic Range LCD TVs”, *IEEE/OSA Journal of Display Technology*, vol. 4, no. 2, pp. 139–146, 2008.

- [36] M. Albrecht, A. Karrenbauer, T. Jung, and C. Xu, “Sorted Sector Covering Combined with Image Condensation: An Efficient Method for Local Dimming of Direct-Lit and Edge-Lit LCDs”, *IEICE Transaction on Electronics*, vol. E93-C, no. 11, pp. 1556–1563, 2010.
- [37] J.-J. Hong, S.-E. Kim, and W.-J. Song, “A Clipping Reduction Algorithm Using Backlight Luminance Compensation for Local Dimming Liquid Crystal Displays”, *IEEE Transactions on Consumer Electronics*, vol. 56, no. 1, pp. 240–246, Feb. 2010.
- [38] S. I. Cho, H.-S. S. Kim, and Y.-H. H. Kim, “Two-step Local Dimming for Image Quality Preservation in LCD Displays”, in *Proceedings of IEEE International SoC Design Conference*, 2011, pp. 274–277.
- [39] *sRGB Color Space*, <http://www.w3.org/Graphics/Color/sRGB/>.
- [40] L. Kerofsky and S. Daly, “Brightness Preservation for LCD Backlight Dimming”, *Journal of the Society of Information Display*, vol. 14, no. 12, p. 1111, 2006.
- [41] H. Nam, “A Color Compensation Algorithm to Avoid Color Distortion in Active Dimming Liquid Crystal Displays”, *IEEE Transactions on Consumer Electronics*, vol. 56, pp. 2569–2576, 2010.
- [42] T. O. Aydın, R. Mantiuk, and H.-P. Seidel, “Extending Quality Metrics to Full Luminance Range Images”, in *Proceedings of SPIE*, 2008, pp. 6806–10.
- [43] J.-R. Ohm, *Multimedia Communication Technology: Representation, Transmission and Identification of Multimedia Signals*. Springer, 2004.
- [44] *SIM2 HDR TV*, <http://www.sim2.com/HDR/>.
- [45] M. Grant and S. Boyd, *CVX: matlab software for disciplined convex programming, version 1.21*, <http://cvx.com/cvx>, Apr. 2011.
- [46] —, “Graph implementations for nonsmooth convex programs”, in *Recent Advances in Learning and Control*, ser. Lecture Notes in Control and Information Sciences, V. Blondel, S. Boyd, and H. Kimura, Eds., Springer-Verlag Limited, 2008, pp. 95–110.

- [47] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C*, 2nd ed. Cambridge University Press, 1992.
- [48] A. Woods, “Understanding Crosstalk in Stereoscopic Displays”, in *3DSA (Three-Dimensional Systems and Applications) Conference (Keynote Presentation)*, May 2010.
- [49] —, “How are Crosstalk and Ghosting defined in the Stereoscopic Literature?”, in *Proceedings of the SPIE*, vol. 7863, 2011, 78630Z.
- [50] D.-S. Kim, Y.-J. Ko, S.-M. Park, J.-H. Jung, and S. Shestak, “Stereoscopic Display Technologies for FHD 3D LCD TV”, in *Proceedings of the SPIE*, vol. 7690, 2010, p. 769 008.
- [51] J.-C. Liou and F.-G. Tseng, “120 Hz low cross-talk stereoscopic display with intelligent LED backlight enabled by multi-dimensional controlling IC”, *Displays*, vol. 30, no. 4-5, pp. 148–154, 2009.
- [52] H. Wang, “Correlations Between Liquid Crystal Director Reorientation and Optical Response Time of a Homeotropic Cell”, *Journal of Applied Physics*, vol. 95, no. 10, p. 5502, 2004.
- [53] P. Adam, P. Bertolino, and F. Lebowsky, “Mathematical Modeling of the LCD Response Time”, *Journal of the Society for Information Display*, vol. 15, no. 8, pp. 571–577, 2007.
- [54] P. Adam, “Améliorations d’artefacts sur panneaux LCD”, PhD thesis, Institut National Polytechnique de Grenoble, 2008.
- [55] I. C. f. D. M. Society for Information Display (SID), *Information display measurements standard, version 1.03*, Jun. 2012.
- [56] *Kodak True Color Image Suite*, <http://r0k.us/graphics/kodak/>.
- [57] A. Olmos and F. A. A. Kingdom, “A biologically inspired algorithm for the recovery of shading and reflectance images”, *Perception*, vol. 33, pp. 1463–1473, 2004.
- [58] *The Consumer Digital Video Library*, <http://www.cdvl.org>.

# Acronyms

**1D** One Dimensional

**2D** Two Dimensional

**3D** Three Dimensional

**AVC** Advanced Video Coding

**CIE** Commission Internationale de l'Éclairage

**CCFL** Cold Cathode Fluorescent Lamp

**CDF** Cumulative Distribution Function

**CIF** Common Intermediate Format

**CRT** Cathode Ray Tube

**DCT** Discrete Cosine Transform

**EEFL** External Electrode Fluorescent Lamp

**FEL** Field Emission Lamp

**FFL** Flat Fluorescent Lamp

**GD** Gradient Descent

**GOP** Group of Pictures

**HCFL** Hot Cathode Fluorescent Lamp

**HD** High Definition

- HDR** High Dynamic Range
- HVS** Human Visual System
- IBL** Initial Backlight Level
- IIR** Infinite Impulse Response
- IPS** In-Plane Switching
- JPEG** Joint Picture Experts Group
- LC** Liquid Crystal
- LCD** Liquid Crystal Display
- LED** Light Emitting Diode
- LUT** Look-Up Table
- MAE** Mean Absolute Error
- MPEG** Motion Picture Experts Group
- MJPEG** Motion Joint Picture Experts Group
- MSE** Mean Squared Error
- OLED** Organic Light Emitting Diode
- PDF** Probability Density Function
- PDP** Plasma Display Panel
- PSF** Point Spreading Function
- PSNR** Peak Signal to Noise Ratio
- PU** Perceptually Uniform
- PWM** Pulse Width Modulation
- RGB** Red, Green and Blue
- SSIM** Structural Similarity

**SROCC** Spearman rank order correlation coefficient

**STD** Standard Deviation

**TN** Twisted Nematic

**TV** Television



Copyright: Nino Burini  
and DTU Fotonik  
All rights reserved  
ISBN: 978-87-93089-03-7

Published by:  
DTU Fotonik  
Department of Photonics Engineering  
Technical University of Denmark  
Ørsteds Plads, building 343  
DK-2800 Kgs. Lyngby

**Nino Burini** was born in Rovigo (Italy) in 1984 and received his B.Sc. and M.Sc. degrees in Computer Science and Engineering from the University of Ferrara (Italy) in 2006 and 2009, respectively. After work and study experiences in Japan and USA, he moved to Denmark in 2010 to start his PhD at DTU in the Department of Photonics Engineering (DTU Fotonik). His research work focused mainly on LCDs with dynamic local backlight, but also touched video post-processing and quality assessment.