Processing Decoded Video for LCD-LED Backlight Display

Post processing of decoded video and local backlight dimming for LCD technology with LED-based backlight

Ehsan Nadernejad

Ph. D. Thesis

September 2013

DTU Fotonik
Department of Photonics Engineering

Processing Decoded Video for LCD-LED Backlight Display

Post processing of decoded video and local backlight dimming for LCD technology with LED-based backlight

Ehsan Nadernejad

September 2013

Technical University of Denmark DTU

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

Abstract

The quality of digital images and video signal on visual media such as TV screens and LCD displays is affected by two main factors; the display technology and compression standards.

Accurate knowledge about the characteristics of display and the video signal can be utilized to develop advanced algorithms for signal (image or video) enhancement. One particular application of such algorithms is the case of LCDs with dynamic local backlight.

The thesis addressed two main problems; first, designing algorithms that improve the visual quality of perceived image and video and reduce power consumption on local LED-LCD backlight. Second, removing the digital video codec artifacts such as blocking and ringing artifacts by post-processing algorithms. A novel algorithm based on image features with optimal balance between visual quality and power consumption was developed. In addition, to remove flickering artifact in LED-LCD display, an adaptive filter was proposed. Several experiments were performed to test and compare the proposed algorithms with the State of the Art algorithms.

Different types of visual artifacts may occur in a video content shown on a display. They can be spatial artifacts such as blocking and ringing or temporal artifacts such as flickering or mosquito. In this thesis, some artifact reduction algorithms for MPEG-2 and H.264/AVC video are proposed.

Resumé

Kvaliteten af digitale billeder og videosignaler på visuelle medier såsom TV- og LCD-skærme bliver hovedsagligt påvirket af to faktorer: displayteknologien og kompressionsstandarder.

Nøjagtig viden om karakteristikken af videosignalets visning kan bruges til at udvikle avancerede algoritmer til signalforbedring (billede eller video). Et anvendt tilfælde af sådanne algoritmer er LCD-skærme med dynamisk lokal baggrundsbelysning.

Denne afhandling består af to hovedproblemer. For det første, at designe en algoritme som forbedrer den opfattede visuelle kvalitet af billeder og video og som reducerer energiforbruget i LED-LCD med lokal baggrundsbelysning. For det andet, at fjerne digitale videokodeksartefakter som "blocking" og "ringing" artefakter ved hjælp af efterbehandlingsalgoritmer. Nye algoritmer baseret på billedkendetegn med optimal balance mellem visuel kvalitet og energiforbrug er blevet udviklet. Derudover er et adaptivt filter til at fjerne flimreartefakter i LED-LCD skærme blevet foreslået. Der er blevet foretaget adskillige eksperimenter for at teste og sammenholde de foreslåede algoritmer med de nyeste konkurrerende algoritmer.

Der kan opstå forskellige typer af visuelle artefakter i en video vist på en skærm. Det kan være spatielle artefakter såsom "blocking" og "ringing" eller temporale artefakter såsom flimren eller "mosquito". I denne afhandling foreslås algoritmer til at reducere artefakter i MPEG-2 og H.264/AVC video.

Acknowledgements

This Ph. D. thesis is the final result of a three year long study in the field of image and video processing for LCD technology.

I have learned a lot about the display world which is changing every day. This project started in the beginning of 2010 and I joined the team October 2010. I am quite impressed by successfully finalizing the work and what I learn during this time. I feel now gratitude, and I wish to thank all the people that have helped me to get here.

I would like to express my deepest gratitude to my supervisor, Prof. Søren Forchhammer, for accepting my application and giving me the opportunity to pursue my Ph.D. at DTU. He has guided me through the project with excellent wisdom, caring and patience, helping me to develop as a researcher and deepen my knowledge, and offered me very useful feedback on my work.

I would like to thank the Danish Strategic Research Council for funding the project and Bang and Olufsen for collaboration. I would like 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 for co-supervision and guiding me patiently, Nino Burini for being my ”Ph. D. brother”, working together and running the experiments and revising this thesis and paper drafts through most of my Ph. D., and Claire Mantel for her contribution in experiments and publications.

I would like to thank my colleagues of the Coding and Visual Communication Group at DTU Fotonik for making nice time to me. A particular thanks to Matteo Salmistraro for all the great time and talks together and Jacob Søgaard for translation of the Danish abstract of this thesis.

I would like to thank again Prof. Zhou Wang for hosting me in his group at Waterloo University in Waterloo, ON (Canada). I have had a great and very productive time there, thanks to him and to my friends Edris Pouresmaeil and Hojatollah Yeganeh.

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

Last, but not least, I would like to thank my wife Sara for her understanding and love during the past few years. Her support and encouragement was in the end what made this dissertation possible. Again, I want to thank Sara one more time for being so kind to review and proofread this thesis.

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.

Ehsan Nadernejad
September 2013
Kongens Lyngby
Denmark

List of Ph. D. Publications

The Ph.D. period has resulted in 21 peer-reviewed publications, including 12 journal papers and 9 conference contributions [1]–[21], listed below.

[1] **E. Nadernejad**, S. Forchhammer, and J. Korhonen, "Artifact Reduction of Compressed Images and Video Combining Adaptive Fuzzy Filtering and Directional Anisotropic Diffusion", in *2011 3rd European Workshop on Visual Information Processing (EUVIP)*, IEEE, 2011, pp. 24–29.

[2] **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.

[3] **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)*, May 2013, pp. 2508–2512.

[4] **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.

[5] **E. Nadernejad**, C. Mantel, N. Burini, and S. Forchhammer, "Flicker Reduction in LED-LCDs with Local Backlight", in *2013 IEEE 15th International Workshop on Multimedia Signal Processing (MMSP)*, IEEE, 2013.

[6] **E. Nadernejad** and S. Forchhammer, “Wavelet-based Image Enhancement Using Fourth Order PDE”, in *2011 IEEE 7th International Symposium on Intelligent Signal Processing (WISP)*, IEEE, 2011, pp. 116–121.

[7] **E. Nadernejad**, S. Forchhammer, and S. Sharifzadeh, “Using Anisotropic Diffusion Equations in Pixon Domain for Image Denoising”, *Signal, Image and Video Processing*, 2012, ISSN: 1863-1703.

[8] **E. Nadernejad**, S. Sharifzadeh, and J. Korhonen, “Improving Performance of Wavelet-based Image Denoising Algorithm Using Complex Diffusion Process”, *Imaging Science Journal*, vol. 60, no. 4, pp. 208–218, 2012.

[9] **E. Nadernejad**, “Improvement of Nonlinear Diffusion Equation Using Relaxed Geometric Mean Filter for Low PSNR Images”, *Electronics Letters*, vol. 49, no. 7, pp. 457–458, 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*, vol. 9, no. 8, pp. 656–665, 2013.

[11] C. Mantel, N. Burini, **E. Nadernejad**, J. Korhonen, S. Forchhammer, and J. Pedersen, “Controlling Power Consumption for Displays With Backlight Dimming”, *Display Technology, Journal of*, vol. 9, no. 12, pp. 933–941, 2013.

[12] N. Burini, C. Mantel, **E. Nadernejad**, J. Korhonen, S. Forchhammer, and J. Pedersen, “Block-Based Gradient Descent for Local Backlight Dimming and Flicker Reduction”, *IEEE/OSA Journal of Display Technology*, vol. PP, no. 99, pp. 1–1, 2013. DOI: 10.1109/JDT.2013.2286213.

[13] 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.

[14] C. Mantel, N. Burini, **E. Nadernejad**, J. Korhonen, and S. Forchhammer, “Quality Assessment of Images Displayed on LCD Screen with Local Backlight Dimming”, in *The Fifth International Workshop on Quality of Multimedia Experience (QoMEX)*, Jul. 2013.

[15] 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.

[16] ——, “Image Dependent Energy-Constrained Local Backlight Dimming”, in *IEEE International Conference on Image Processing (ICIP)*, Sep. 2012, pp. 2797–2800.

[17] **E. Nadernejad** and M. Nikpour, “Image Denoising Using New Pixon Representation Based on Fuzzy Filtering and Partial Differential Equations”, *Digital Signal Processing*, vol. 22, no. 6, pp. 913–922, 2012.

[18] **E. Nadernejad** and A. Barari, “A Novel Pixon-Based Image Segmentation Process Using Fuzzy Filtering and Fuzzy C-mean Algorithm”, *International Journal of Fuzzy Systems*, vol. 13, no. 4, pp. 350–357, 2011.

[19] **E. Nadernejad** and S. Sharifzadeh, “A New Method for Image Segmentation Based on Fuzzy C-means Algorithm on Pixonal Images Formed by Bilateral Filtering”, *Signal, Image and Video Processing*, vol. 7, no. 5, pp. 855–863, 2013.

[20] **E. Nadernejad**, H. Hassanpour, and H. Naimi, “High-speed MRF-based Segmentation Algorithm Using Pixonal Images”, *Imaging Science Journal*, vol. 61, no. 7, pp. 592–600, 2013.

[21] M. Salarian, **E. Nadernejad**, and H. MiarNaimi, “A New Modified Fast Fractal Image Compression Algorithm”, *Imaging Science Journal*, vol. 61, no. 2, pp. 219–231, 2013.

Chapters 2 and 3 of this thesis are based on 5 of these publications [1]–[5], reported in Appendix A. Chapter 2 is based on [1]–[3] with a submitted journal paper, while Chapter 3 is based on [4], [5].

Contents

1 Introduction **1**

1.1 Motivation 1

1.1.1 Post-processing for Artifact Removal 2

1.1.2 Dynamic Backlight Dimming in LCD 2

1.2 Main Contributions 3

1.3 Goals and Structure of the Thesis 4

2 Image and Video Post-Processing **13**

2.1 Image and Video Compression and Distortion 14

2.2 State of the Art 16

2.3 Artifact Removal by Anisotropic Diffusion and Fuzzy Filtering 18

2.3.1 Anisotropic Diffusion 18

2.3.2 Fuzzy Filter 20

2.3.3 Deblocking and Deringing Method 21

2.3.4 Artifact Reduction in H.264/AVC Video Sequences 24

2.3.5 Experiments and Results 25

2.3.5.1 Compressed Images 25

2.3.5.2 Video Sequences 26

2.3.5.3 Subjective Evaluation 27

2.4 Adaptive Deblocking and Deringing of H.264/AVC video . 34

2.4.1 Deblocking Step 34

2.4.2 Bilateral Filter for Deringing 38

2.4.3 Experiments and Results 39

2.5 Post-Processing of Video Sequences Using Deblocking and Adaptive Trilateral Filtering 41

2.5.1 Trilateral Filter for Removing Ringing Artifact . . 42

2.5.2 Experiments and Results 45
2.6 Conclusion 54

3 LCD Backlight Dimming 55
3.1 LED-LCD Backlight Display 55
3.1.1 Background of Backlight Display 56
3.1.2 State of the Art Dimming Algorithms 60
3.1.2.1 Feature-Based Algorithms 60
3.1.2.2 Optimization-Based Algorithms 69
3.1.3 Modeling 73
3.1.3.1 Transmittance and leakage 73
3.1.3.2 Backlight diffusion 75
3.1.3.3 Brightness compensation 75
3.1.3.4 Quality Assessment of Algorithms 78
3.1.3.5 Equipment 78
3.1.4 Local Feature-Based Algorithm 79
3.1.4.1 Experiments 81
3.1.4.2 Objective Evaluations 81
3.1.4.3 Subjective Evaluations 85
3.2 Flicker Reduction in LED-LCDs with Local Backlight . . 88
3.2.1 State of the Art Flicker Reduction Algorithms . . 88
3.2.1.1 Lee Algorithm 89
3.2.1.2 Chen Algorithm 90
3.2.2 Flicker Reduction Using IIR Filter 91
3.2.2.1 Experiments 93
3.3 Conclusion 101

4 Conclusion and Future Work 103

Appendix A Ph. D. Publications 109
A.1 Artifact Reduction of Compressed Images and Video Combining Adaptive Fuzzy Filtering and directional Anisotropic Diffusion 111
A.2 Enhancing Perceived Quality of Compressed Images and Video With Anisotropic and Fuzzy Filtering 119
A.3 Adaptive De-Blocking and De-Ringing of H.264/AVC Video Sequences 141

A.4 Adaptive Local Backlight Dimming Algorithm Based on Local Histogram and Image Characteristics 149
A.5 Flicker Reduction in Led-Lcds with Local Backlight . . . 165
A.6 Wavelet-based Image Enhancement using Fourth Order PDE 173
A.7 Using Anisotropic Diffusion Equations in Pixon Domain for Image Denoising 181
A.8 Improving Performance of Wavelet-based Image Denoising Algorithm using Complex Diffusion Process 195
A.9 An Improvement of Nonlinear Diffusion Equation using Relaxed Geometric Mean Filter for Low PSNR Images . . 209
A.10 Modeling Power-Constrained Optimal Backlight Dimming for Color Displays 213
A.11 Controlling Power Consumption for Displays With Backlight Dimming 225
A.12 Block-Based Gradient Descent for Local Backlight Dimming and Flicker Reduction 237
A.13 How to Evaluate Objective Video Quality Metrics Reliably 249
A.14 Quality Assessment of Images Displayed on LCD Screen with Local Backlight Dimming 257

Appendix B Testsets 261

List of Figures

2.1 Flow chart for image filtering method 21

2.2 Fuzzy filter VS anisotropic diffusion 22

2.3 Flow chart for spread parameter selection 23

2.4 H.264/AVC block boundary 24

2.5 1D anisotropic diffusion VS 2D anisotropic diffusion . . . 24

2.6 Flow chart for video filtering method 25

2.7 Visual comparison of filters on JPEG 29

2.8 Visual comparison of filters on MJPEG 30

2.9 PSNR comparison of filtering methods 31

2.10 MSSIM comparison of filtering methods 31

2.11 Visual comparison of filters on H.264/AVC 32

2.12 PSNR comparison of filtering methods for H.264/AVC sequence 33

2.13 MSSIM comparison of filtering methods for H.264/AVC sequence 33

2.14 Flowchart of artifact removal method 35

2.15 Relation between MSE and quantization step 36

2.16 Decision mode pixel vector and filtering window 37

2.17 Decision modes partitioning 37

2.18 In-loop filter comparison 41

2.19 Texture map detection 44

2.20 PSNR comparison for Foreman sequence 46

2.21 MSSIM comparison for Foreman sequence 47

2.22 Visual comparison of algorithms applied on *Foreman* sequence (Frame No.56) 50

2.23 Different bit-rates comparison 52

2.24 Visual Difference between two algorithms 53

3.1 Direct and Edge Backlight 57
3.2 Dimming Methods 59
3.3 Cho Algorithm (first algorithm) 62
3.4 Cho algorithm (second algorithm) 63
3.5 Chen Algorithm 66
3.6 Wang Algorithm 67
3.7 Lookup Table of Wang Algorithm 68
3.8 Kang and S.I.Cho Algorithms 70
3.9 Albrecht algorithm 71
3.10 Generic local backlight dimming system 74
3.11 Example of modeled and measured backlight 76
3.12 Histogram percentile 80
3.13 MSE performance, histogram based algorithm on edge screen 83
3.14 MSE performance, histogram based algorithm on SIM2 screen 84
3.15 PSNR performance, histogram based algorithm on edge screen 84
3.16 PSNR performance, histogram based algorithm on SIM2 screen 85
3.17 Example of the flickering artifact 91
3.18 Proposed flicker reduction algorithm 92
3.19 Test video sequences 94
3.20 Performance of flicker reduction algorithm on Edge-lit display 96
3.21 The average LED values before and after filtering 97
3.22 PSNR before and after filtering 97
3.23 MSE before and after filtering 98
3.24 The average PSNR vs. power consumption 98

B.1 “ICIP” testset 262
B.2 “ICIP” testset, grayscale 263
B.3 Kodak True Color testset 264
B.4 Kodak True Color testset, grayscale 265
B.5 Four images for experiments 266
B.6 Video testset 267

List of Tables

2.1 Subjective ranking on MJPEG post-processing methods . 28

2.2 Subjective ranking on H.264/AVC post-processing methods 28

2.3 H.264/AVC post-processing performance 40

2.4 Selected W-PSNR results for all sequences 40

2.5 Performance related to the in-loop filter 41

2.6 Y-PSNR, W-PSNR and Y-MSSIM of the proposed algorithm compared with State of the Art algorithms 48

2.7 Selected Y-PSNR results for all sequences 49

2.8 SSD performance of different algorithms 49

2.9 Selected Y-SSD results for all sequences 51

3.1 Subjective rankings, histogram based algorithm 87

3.2 The average de-flickering results for different algorithms on all video sequences 99

3.3 Performance of proposed algorithm using SSD measurement 100

Chapter 1

Introduction

1.1 Motivation

Digital images and video is a part of our everyday life. Video, being an image sequence, is often called video sequence as well.

Everyday, numerous digital images and video contents are captured, coded, stored, transmitted, decoded and displayed all over world. From capture to display, many factors affect the final perceived quality. In many applications video sequences are compressed using different standards such as MPEG-2 or H.264/AVC. Video compression is performed by removing redundancy from a signal. Most of the practical video compression methods like H.264/AVC are based on lossy compression, thus the compression will introduce a penalty so that the decoded video is not identical to the original signal. As a result, the decoded signal noticeably represents lower quality compared with the original signal due to different artifacts. The artifacts may be blocking and ringing in spatial domain and mosquito and flickering in time domain. The spatial artifacts are the same as the compressed image and video signal, while the temporal artifacts appear only in video signal. The artifacts are more annoying and the video quality get worse when the coding bit rate decreases.

Post-processing algorithms can improve the quality of decoded image or video signal. There are different algorithms for post-processing and artifact reduction of decoded signals.

The final perceived quality is also significantly affected by the display technology. This occurs after the post processing step in the visualization device. There are varieties of displays in different sizes and contexts of use, e.g. a mobile cell phone or a Television (TV).

In digital TV, post-processing algorithms are used for improving the visual quality of the decoded signal. The Liquid Crystal Display (LCD) is used in a wide variety of devices, including TV sets and computer monitors. In the case of LCDs, LED backlight luminance can be varied adaptively to improve the contrast and reduce the power consumption.

1.1.1 Post-processing for Artifact Removal

Video compression reduces the number of bits necessary to store or transmit a signal. However, it can introduce visual artifacts, particularly when the compression is aggressive. The two of the most common artifacts are *blocking*, caused by the inherently block based structure of many video codecs (including MPEG-2 and H.264/Advanced Video Coding (AVC)), and *ringing*, caused by the domain conversions and data quantization, adopted by many modern video standards. The temporal variations of these artifacts causes other artifacts, i.e. mosquito noise and flickering. All of these artifacts reduce the visual quality of the coded video, and many methods have been proposed to combat them.

1.1.2 Dynamic Backlight Dimming in LCD

LCD is nowadays the most widespread display type. That is used in 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 [22]. LED-based backlights can be divided in independently adjustable segments, called *local* backlights. This is 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 segment can be dimmed to match the brightness of the respective image area. This saves power, since dark areas require less light.

Local 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.

1.2 Main Contributions

This Ph.D. thesis focuses on post-processing and local backlight dimming of multidimensional signals for LED-LCD backlight display. The first part of this Ph.D. project is dedicated to post-processing of compressed images and video using codecs based on block-based transformations. Three novel algorithms, based on block classification and adaptive combination of existing techniques like anisotropic diffusion, fuzzy and bilateral filters, were proposed to reduce blocking and ringing artifacts; their superior performance was evaluated with objective and subjective methods. In the first algorithm, an adaptive method was proposed to remove blocking and ringing artifacts in compressed images and videos. The experimental results showed that the algorithm could also remove the temporal artifacts observed subjectively and objectively. The second and third algorithms used a quality metric for I, P, B frames to remove blocking artifacts. For removing the ringing artifacts, an adaptive bilateral filter was proposed in the second algorithm. In addition, a trilateral filter was proposed in the third algorithm in which a texture detection method was used to calculate the new term of the trilateral filter. Finally, the State of the Art algorithms were implemented and compared with the proposed algorithms.

The second part of this Ph.D. project involved working on local backlight dimming. An initial model of backlight dimming systems was developed [16]. Most State of the Art backlight dimming algorithms were implemented, and then, a new feature-based algorithm was proposed [4]. The proposed method has been applied on two modeled screens: one with a high resolution direct-lit backlight, and the other screen with 16 edge-lit backlight segments placed in two columns and eight rows. The results show that the proposed algorithm provides better trade-off between power consumption and image quality preservation than the other tested algorithms. In addition, to study the performance of the proposed algorithm, both objective and subjective ratings were performed [4]. For removing flickering artifact in LED-LCD backlight, an adpative second order IIR filter was proposed [5]. it uses the target image features to find appropriate LED value and reduce the flicker.

1.3 Goals and Structure of the Thesis

This Ph.D. period has resulted in 21 peer-reviewed publications, including 12 journal publications and 9 conference contributions [1]–[21]. The publications are in the fields of display technology, mathematical modeling, image processing, video coding, visual quality assessment and information theory.

Publications [18]–[20] are in the field of image segmentation and tracking and publication [21] is in the fractal coding domain. In [6]–[9], [17] algorithms for image denoising and enhancement are presented. These algorithms are useful for improving quality of digital image and video. For reducing the coding artifacts, some new algorithms are introduced in publications [1]–[3]. In the field of display technology, publications [4], [5], [10]–[12], [15], [16] are presented some new feature-based and optimization-based algorithms for LED-LCD backlight display. Assessment of images and video displayed on LCD screen are explained in [13], [14]. This thesis comprises 14 publications [1]–[14]. They are included in Appendix A, and also briefly described at the end of this Section. The main technical content reported in Chapters 2 and 3 is centered around publications [1]–[5] which are in Appendix A.

Chapter 2 discusses post-processing of images and video compressed with block-based codecs. These codecs can create visual artifacts such as blocking and ringing as well as temporal artifacts such as mosquito noise and flickering. After presenting the State of the Art algorithms, three deblocking and deringing methods (based on fuzzy filtering, anisotropic diffusion, decision mode, bilateral and trilateral filtering) are presented.

Chapter 3 presents local backlight dimming. After a brief introduction to LCD, the benefits and challenges of the technique are discussed, followed by the description of State of the Art algorithms. A model for local backlight dimming LCD [10] is introduced that includes important aspects like light diffusion and perception. Then, it is used as a basis for dimming 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. At the end, an adaptive algorithm for removing flickering artifact in LED-LCD backlight is presented.

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

Brief Description of the Publications

1. **Artifact Reduction of Compressed Images and Video Combining Adaptive Fuzzy Filtering and Directional Anisotropic Diffusion [1]:**

   In this paper, we have proposed an effective algorithm for image and video artifact removal using an adaptive fuzzy filter and directional anisotropic diffusion. This novel method overcomes the limitations of the conventional nonlinear filters by taking pixel's activity and the direction between pixels both into account. It has been shown that the proposed algorithm improves the visual quality of compressed images and videos in terms of PSNR and MSSIM, compared to existing approaches.

2. **Enhancing Perceived Quality of Compressed Images and Video with Anisotropic Diffusion and Fuzzy Filtering [2]:**

   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.

3. **Adaptive Deblocking and Deringing of H.264-AVC Video Sequences [3]:**

   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.

4. **Adaptive Local Backlight Dimming Algorithm Based on Local Histogram and Image Characteristics [4]:**

   This paper introduces a local backlight dimming algorithm using local histograms to determine the backlight level. The algorithm consists of three steps. 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.

5. **Flicker Reduction in LED-LCDs with Local Backlight [5]:**

   Fast variations of backlight dimming over time can produce the visually annoying artifact called flickering. To avoid the flickering artifact we introduce an adaptive second order Infinite Impulse Response filter. The filter uses spatial and temporal information of LED values. The coefficients of the IIR were selected adaptively using local features of the target image. To show the performance of the proposed algorithm, some State of the Art local backlight dimming algorithms were implemented and tested on 8 Full-HD video sequences with different features. Experimental results demonstrated that the proposed algorithm can avoid abrupt variations of the LED signals and consequently reduce the flicker artifact.

6. **Wavelet-based image enhancement using fourth order PDE [6]:**

   Wavelet thresholding is used in many image de-noising techniques. In these approaches the thresholding is not applied on the approximate sub-bands, which have the main information about the signals. Since noise affects both the approximate and detail subbands of images, the de-noised images using these approaches have restriction in improvement. In this study, a new approach using the combination of wavelet thresholding and fourth order partial differential equations was introduced for image de-noising. In the proposed method, both thresholding and fourth order diffusion methods are combined to reduce the noise in all subbands of the image. Results obtained on a number of standard images illustrate that the proposed method presents a better performance compared to the existing wavelet thresholding method in image de-noising.

7. **Using Anisotropic Diffusion Equations in Pixon Domain for Image De-noising [7]:**

   Image enhancement is an essential phase in many image processing algorithms. In any image de-noising algorithm, it is a major concern to preserve the interesting structures of the image. Such interesting structures in an image often correspond to the discontinuities in the image (edges). In this paper, we propose a new algorithm for image de-noising using anisotropic diffusion equations in pixon domain. In this approach, diffusion equations are applied on the pixonal model of the image. The algorithm has been examined on a variety of standard images and the performance has been compared with algorithms known from the literature. The experimental results show that in comparison with the other existing methods, the proposed algorithm has a better performance in de-noising and preserving image edges.

8. **Improving Performance of Wavelet-based Image De-noising Algorithm using Complex Diffusion Process [8]:**

   Image enhancement and de-noising is an essential pre-processing step in many image processing algorithms. In any image de-noising algorithm, the main concern is to keep the interesting structures of the image. Such interesting structures often correspond to the discontinuities (edges). In this paper, we present a new algorithm for image noise reduction based on the combination of complex diffusion process and wavelet thresholding. In the existing wavelet thresholding methods, the noise reduction is limited, because the approximate coefficients containing the main information of the image are kept unchanged. Since noise affects both the approximate and detail coefficients, the proposed algorithm for noise reduction applies the complex diffusion process on the approximation band in order to alleviate the deficiency of the existing wavelet thresholding methods. The algorithm has been examined using a variety of standard images and its performance has been compared against several de-noising algorithms known from the prior art. Experimental results show that the proposed algorithm preserves the edges better and in most cases, improves the measured visual quality of the denoised images in comparison to the existing methods known from the literature. The improvement is obtained without excessive computational cost, and the algorithm works well on a wide range of different types of noise.

9. **Improvement of Nonlinear Diffusion Equation Using Relaxed Geometric Mean Filter for Low PSNR Images [9]:**

   A new method to improve the performance of low PSNR image de-noising is presented. The proposed scheme estimates the edge gradient from an image that is regularized with a relaxed geometric mean filter. The proposed method consists of two stages; the first stage consists of a second order nonlinear anisotropic diffusion equation with new neighboring structure and the second is a relaxed geometric mean filter, which processes the output of nonlinear anisotropic diffusion equation. The proposed algorithm enjoys the benefit of both nonlinear PDE and relaxed geometric mean filter. In addition, the algorithm will not introduce any artifacts, and preserves image details, sharp corners, curved structures and thin lines. Comparison of the results obtained by the proposed method, with those of other methods, shows that a noticeable improvement in the quality of the de-noised images, that were evaluated subjectively and quantitatively, is produced.

10. **Modeling Power-Constrained Optimal Backlight Dimming for Color Displays [10]:**

    In this paper, we present a framework for modeling color liquid crystal displays (LCDs) having local light-emitting diode (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.

11. **Controlling Power Consumption for Displays With Backlight Dimming [11]:**

    This paper presents a global framework for a backlight power management system. High-performance and low-complexity approaches are then introduced, both providing means to achieve any target power level. Compared to algorithms from the literature, the high-performance method achieves the best possible quality for a given power consumption and the low-complexity method achieves a good trade-off between power and quality. Finally, the high-performance algorithm is extended to video in three modes [11].

12. **Block-Based Gradient Descent for Local Backlight Dimming and Flicker Reduction [12]:**

    This paper presents a fast version of an iterative backlight dimming algorithm based on gradient descent search. The speed is increased by exploiting the local uniformity of backlight to reduce the complexity for determining the optimal backlight of a given image. The algorithm is adapted to video sequences by using the backlight of the previous frame as a starting point to calculate the backlight for the current frame. Fast variations of backlight dimming over time can produce a visually annoying flicker artifact. To avoid this artifact, a control mechanism is also proposed.

13. **How to Evaluate Objective Video Quality Metrics Reliably [13]:**

    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) [23]. 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.

14. **Quality Assessment of Images Displayed on LCD Screen with Local Backlight Dimming [14]:**

    This paper presents a subjective experiment collecting quality assessment of images displayed on a LCD with local backlight dimming using two methodologies: absolute category ratings and paired-comparison. Some well-known objective quality metrics are then applied to the stimuli and their respective performance are analyzed.

Chapter 2

Image and Video Post-Processing

This Chapter, is dedicated to the post-processing algorithms. They can improve the quality of compressed images and videos. This Chapter is based on papers [1]–[3] which are also reported in Appendix A.

Image and video signals is compressed, or *encoded*, to reduce the number of bits needed for their storage or transmission. When needed, the signal can be decompressed, or *decoded*. If the decoded signal is identical to the original signal (before coding), then the compression is *lossless*, because no information is lost. Otherwise, it is *lossy*. The compression rates (the ratio of the number of bits used to represent the original signal and that used to represent the coded signal) achieved by lossy coding are typically higher than lossless coding, at the cost of reduced fidelity. Lossless coding exploits the fact that images and video signals typically contain redundancies that can be removed upon coding and reconstructed upon decoding. Lossy compression typically removes information that has limited impact or no impact on visual perception. However, to achieve high compression rates, the information removal might be so aggressive that it creates visible *coding artifacts*. Therefore, compression rate and visual quality (in terms of absence of artifacts) are in a trade-off relationship. When aggressive compression is required, the artifacts can seriously degrade the visual quality of the signal. The severity of the artifacts can be reduced by *post-processing* the decoded signal. Utilizing of the coding techniques allows to design effective post-processing algorithms that can improve the visual quality of lossy signals.

This Chapter includes, in Section 2.1 a description of some aspect of block-based lossy coding (e.g. JPEG images, MJPEG and H.264/AVC videos), followed by the presentation of the State of the Art algorithms in Section 2.2. Finally, three relevant post-processing methods are presented in Section 2.3 and 2.4, respectively.

2.1 Image and Video Compression and Distortion

Many lossy image and video codecs are block based and exploit the Discrete Cosine Transform (DCT) or some variations. The images initially in the Red, Green and Blue (RGB) color space are converted to the $\mathrm{YC_bC_r}$/YUV color spaces, where compression is typically performed.

In the Joint Photographic Experts Group (JPEG) image compression standard, the image is divided in blocks of 8×8 pixels. Each block is then converted from the spatial domain to the frequency domain using the DCT transform; the pixel values are thus transformed into coefficients that represent the "frequencies" composing the block. Typically, the low frequency coefficients represent the overall characteristics of the block, while the high frequency coefficients contain information about the details of the image block, e.g. sharp edges or textures. The first coefficient is called DC coefficient and contains information about the average pixel value in the block; the remaining 63 are called AC coefficients. The low frequency coefficients tend to have higher magnitude than the high frequency ones. Compression is achieved by *quantizing* the coefficients, that is dividing them by an integer (q factor) and rounding the result to the closest integer. The JPEG standard specifies a quantization matrix that assigns a specific quantizer to each coefficient. Typically, after quantization low frequency coefficients have the largest values, while most high frequency coefficients become zero or small numbers, which require less bits to be encoded efficiently. The magnitude of the quantizers is an encoding parameter defined in the standard. Larger quantizers result in higher compression rates, but also more loss of information (and consequently worse visual quality). When decoding, the quantized coefficients are multiplied by the quantizers specified in the quantization matrix, then an inverse DCT is applied to convert from the transform domain to the spatial domain. The resulting pixel values approximate the original ones with an accuracy proportional to the magnitude of the quantizers. More information on the JPEG standard can be found in [24]. JPEG images are also used to save frames in the Motion Joint Picture Experts Group (MJPEG) standard, which is simply a sequence of JPEG images.

Several video compression standards are also based on blocks and DCT, like MPEG-2 and its successor H.264/MPEG-4 Advanced Video Coding (AVC). Differently from static images, videos are sequences of frames. Usually, a frame is similar to the previous and next frames. This temporal redundancy can be exploited to compress the video stream. There are three type of coded frames: I frames, P frames and B frames. The I frames can be decoded independently from other frames. In MPEG-2 they are coded similar to the JPEG images. In H.264/MPEG-4 AVC the additional intra prediction step predicts an approximate version of the frame, which is subtracted from the original frame; the residuals of the subtraction are then coded by an adaptive transform. In H.264/MPEG-4 AVC, the transform blocks can have different sizes, i.e. 4×4 or 8×8 pixels). The P frames use preceding I frames or P frames as references to reconstruct an approximated version of themselves. For this aim, a search is performed for each macro block, to find its corresponding macro block in the reference frame based on a similarity measure [25], [26]. A so called *motion vector* specifying the offset of the macroblock in the reference frame is saved for each macroblock and used, when decoding, to reconstruct the predicted P frame. As for I frames, the residuals between the original and predicted frames are transformed and coded. In H.264/MPEG-4 AVC, the macroblock is typically divided in blocks of variable sizes (from 4×4 to 16×16 pixels) and a motion vector is found for each block. B frames are predicted like P frames, but can use a following frame as reference. I, P and B frames are typically organized in a Group of Pictures (GOP), that is a structure defining in what sequence the frames are coded and displayed. For instance, a GOP can have 12 frames in the following sequence: IBBPBBPBBPBB. This can also be described with the notation $(\mathrm{IBBP})_{12}$. More information on the MPEG-2 and H.264/MPEG-4 AVC standards can be found in [25] and [26].

These codecs can compress images and video. However this may degrade the image quality and introduce visual artifacts. The separate compression of each block can generate *blocking* artifacts, which creates visible horizontal and vertical edges at the block borders. Quantization or removal of the high-frequency DCT coefficients can cause ripples or oscillations around sharp edges and contours, also known as *ringing*. The temporal variation of ringing in the same block generates *mosquito* noise in video. Similarly, temporal quality inconsistencies in the same block create *flickering*.

These artifacts might not cause significant measurable distortion, but the Human Visual System (HVS) has shown to be more sensitive to them. It is then important to design algorithms for artifact removal to improve the visual quality of coded image and video. The H.264/AVC standard includes an in-loop filter for deblocking, that helps improving the visual appearance of coded frames.

2.2 State of the Art

There are many algorithms combating spatial and temporal artifacts. Zhai proposed a deblocking algorithm [27], consisting of regularization of AC coefficients in shifted DCT blocks, block-wise shape adaptive filtering in the spatial domain, and DCT quantization constraining. Kim [28] proposed an adaptive deblocking algorithm for low bit-rate video coding, where blocks are categorized as low or high activity depending on their DC and AC values; different adaptive low-pass filters are then applied depending on the category. Ren et al. [29] proposed a new algorithm for both the flickering and blocking artifacts reduction. For reducing the flickering effect between adjacent frames, an adaptive multi-scale motion filtering method to maintain the motion coherence of processed video was proposed. For blocking artifacts suppression, it adopts a patch-based scheme, in which similar patches are grouped in a spatio-temporal domain and each patch group is recovered by solving a low rank matrix completion problem. In [30], patch clustering and low-rank minimization is used to reduce blocking artifacts. First, the whole compressed image is divided into small patches. For each patch, the algorithm performs patch clustering to collect similar patches into a group. Then, the whole group is simultaneously reconstructed by a low-rank minimization approach. Chen et al. [31] introduced three filtering modes using the activity across block boundary to remove blocking effect. By considering the masking effect of the HVS (Human Visual System), an adaptive filtering decision was integrated into the deblocking process [31]. Yeh et al. [32] presented a low-complexity post-processing deblocking filter which is suitable for various video standards. The proposed algorithm automatically detects blocking effects at 4×4 or 8×8 block boundaries. In the filtering stage, the algorithm provides four filter modes to eliminate the effects of blocking in various frequency regions based on region activity analysis. In addition, the algorithm also considers the chrominance filter to improve quality. A novel fusion based technique was proposed to minimize the loss of useful information which may result from the deblocking operation by Saleem et al. [33]. In [34] Francisco et al. proposed a post-processing deblocking technique that is independent of the compression method used to encode the image. It is based on the use of a space-variant finite impulse response (FIR) filter, with an adaptive number of coefficients. The algorithm [34] performs a total variation analysis of the encoded images to build a map that defines the filter's shape and length for each region on the image. Regions with low total variation are assumed to have a smooth nature and are strongly filtered, using filters with wide support regions to eliminate the artifacts in the block's boundaries. Regions with high total variation are assumed to contain a high level of detail, and are only softly filtered. Linear low-pass filters have been used in [35], [36]. Deblocking algorithms based on anisotropic diffusion were presented in [37]–[39]. Often, deblocking methods introduce undesired blurriness.

Ringing is reduced in [40], [41] using linear or non-linear isotropic filters near edge regions. Flickering reduction methods focus on intra-frame coding, where it is more evident. In [42], [43], flickering is reduced based on the quantization error and on choosing the optimal intra prediction mode.

In [41], [44]–[46], fuzzy filters are used for artifact removal. For example in [45] an adaptive fuzzy filter was proposed to reduce the artifacts in compressed images and video. For JPEG images, a fuzzy spatial filter was used based on the directional characteristics of ringing artifacts along the strong edges [45]. For compressed video sequences, the motion compensated spatio-temporal filter (MCSTF) was applied to intraframe and interframe pixels to deal with the both spatial and temporal artifacts [45]. There are also more methods that can be found [47]–[52].

2.3 Artifact Removal by Anisotropic Diffusion and Fuzzy Filtering

This section presents two techniques to remove blocking and ringing artifacts through the combined use of anisotropic diffusion equations and fuzzy filtering [1], [2]. The first proposed method was designed for still images, and the second was for video sequences. Fuzzy filtering has proven to be 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 were designed to be adaptive to the image and video content.

The performance of the proposed methods was compared against other methods by using different objective quality metrics and a subjective comparison study. The results show that the proposed algorithms that employ directional and spatial fuzzy filters performed better artifact reduction than other methods, on both still images and video sequences, even for H.264/AVC compressed video. The algorithms were first presented in [1] and then were extended in [2]

2.3.1 Anisotropic Diffusion

Orientation and frequency are fundamental spatial information processed by the HVS [53] that need to be taken into account in image processing algorithms. The anisotropic diffusion, also called Perona-Malik diffusion, was first proposed in [54]. It is a mathematical model that has found many applications in signal and image processing for removing noise while keeping important details like sharp edges, lines or texture.

Let a gray scale image $I(x, y)$ be represented by a real valued mapping $I : \mathbb{R}^2 \Rightarrow \mathbb{R}$. The initial image I_0 is modified through the anisotropic diffusion equation:

$$\frac{\partial I}{\partial t} = \mathrm{div}[c(\|\nabla I\|)\nabla I] \tag{2.1}$$

$$I(x, y, t)|_{t=0} = I_0(x, y), t \in (0, T), \tag{2.2}$$

where div denotes the divergence operator, ∇ is the gradient operator, $\|\nabla I\|$ is the gradient magnitude of I, $I(x, y, t)$ is the image at position (x, y) at time t (where t is a scale parameter used to enumerate iteration steps), and $c(\|\nabla I\|)$ is a coefficient controlling the diffusion speed. The diffusion coefficient $c(\|\nabla I\|)$ decreases monotonically as the magnitude of the gradient increases. As the highest values of the gradient are perpendicular to the edges, it smooths intra-region but not inter-region by impeding diffusion across edges. If $c(\|\nabla I\|)$ varies according to the local image gradient, there is anisotropic diffusion. Here the function is

$$c(x, y, t) = \exp\left(-\frac{\|\nabla(I(x, y, t))\|^2}{k^2}\right), \tag{2.3}$$

as in [54], where k is a parameter used to differentiate edge and smooth regions. The value of k is calculated from the variance of the magnitude of the gradient:

$$k = \phi \times \text{var}(|\nabla I|), \tag{2.4}$$

where ϕ is a constant equal to 2×10^{-6} [1]. To avoid inaccuracies due to noise, it has been suggested to calculate the gradient on a smoothed version of the image, obtained with a spatial Gaussian filter G_σ.

The anisotropic diffusion iterations can be explicited solving Eq. 2.1, approximating spatial derivatives with central differences and the temporal derivative with the forward difference:

$$I_{i,j}^{n+1} = I_{i,j}^{n} + \frac{\Delta t}{|\eta_{i,j}|} \sum_{p \in \eta_{i,j}} c\left((\|\nabla I_{i,j}^{n}\|)_p\right) \cdot \left((\nabla I_{i,j}^{n})_p\right) \tag{2.5}$$

where $I_{i,j}^{n}$ is the discretely sampled image at pixel position (i, j), $\eta_{i,j}$ is the spatial neighborhood of pixel (i, j), $|\eta_{i,j}|$ is the cardinality of $\eta_{i,j}$, c is the diffusion coefficient of Eq. 2.3, n is the iteration counter Δt is the time step size.

In this work, video frames are processed in the YUV format, each channel separately.

2.3.2 Fuzzy Filter

Fuzzy filter has been applied to coding artifacts reduction recently [41], [43]–[45]. Fuzzy filters process pixels depending on the values of the pixels in their neighborhood. The general normalized output $\hat{I}[i,j]$ for a pixel at $(i,j) \in \mathbb{Z}^2$ with a set ω of neighboring pixels is:

$$\hat{I}[i,j] = \frac{\sum_{[i',j'] \in \omega} h(I[i+i',j+j'], I[i,j]) I[i+i',j+j']}{\sum_{[i',j'] \in \omega} h(I[i+i',j+j'], I[i,j])} \tag{2.6}$$

where $h(I[i+i',j+j'], I[i,j])$ is a filter, also called membership function [41], [46], determining the weight of the input pixel intensities $I[i+i',j+j']$. The challenge of the membership function is to remove artifacts smoothing flat regions, as a low-pass filter, without blurring detailed areas.

In this work, the membership function is Gaussian as in [44], [45]:

$$h(I[i+i',j+j'], I[i,j]) = \exp\left(-\frac{(I[i+i',j+j'] - I[i,j])^2}{2\sigma^2}\right), \tag{2.7}$$

where σ represents the spread parameter of the input and can be varied depending on i, i', j, and j' to control the strength of the fuzzy filter. The input $I[i,j]$ always contributes to the output equally or more than the other samples. Larger values of σ mean a stronger influence of neighboring pixels. An adaptive spread parameter can change the effect of the filter depending on the activity level of the area (smooth or detailed):

$$\sigma(I[i+i',j+j'], I[i,j]) = K[i+i',j+j'] \times \sigma_A[i,j], \tag{2.8}$$

where σ_A is a amplitude function of the dependent on the position and K is a scaling function dependent on the direction of $[i+i',j+j']$ to $[i,j]$. In this work $K = 1$, as in [38]. A cosine-based general form for the spread parameter was used in [44]:

$$\sigma(\theta) = \sigma_A(\alpha + \beta \cos^2(\theta)), \tag{2.9}$$

where σ_A is a function of the standard deviation of pixel intensities [38], θ is the angle between pixel $[i,j]$ and $[i+i',j+j']$ relative to the horizontal direction, and α and β are positive scaling factors controlling the maximum and minimum strength of the filter [39].

More details about the membership function and fuzzy filtering in general are available in [39], [41], [44], [45]. The definitions in this section are for Two Dimensional (2D) fuzzy filtering; One Dimensional (1D) fuzzy filters can be derived from Eqs. 2.6- 2.8 by removing one of the dimensions.

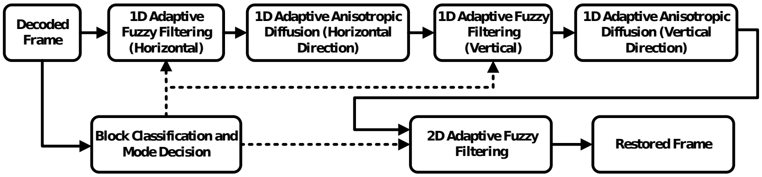

Decoded Frame
1D Adaptive Fuzzy Filtering (Horizontal)
1D Adaptive Anisotropic Diffusion (Horizontal Direction)
1D Adaptive Fuzzy Filtering (Vertical)
1D Adaptive Anisotropic Diffusion (Vertical Direction)
Block Classification and Mode Decision
2D Adaptive Fuzzy Filtering
Restored Frame

Figure 2.1: Flowchart of the proposed method for image filtering. Solid lines denote the data flow and dashed lines the parameter flow [2].

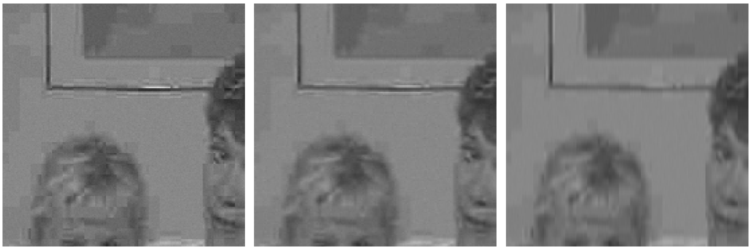


Figure 2.2: Effects of removing the adaptive fuzzy filter and adaptive anisotropic diffusion: the original compressed image (left), 1D adaptive fuzzy filtering (both directions, center) and 1D adaptive anisotropic diffusion (both directions, right) [2].

2.3.3 Deblocking and Deringing Method

The strengths of fuzzy filters and anisotropic diffusion are combined to combat deblocking and ringing artifacts. While 1D fuzzy filtering can remove blocking, anisotropic diffusion is better at removing ringing noise, as shown in Figure 2.2.

Figure 2.1 shows the flow chart of the method being presented. As first step, the image blocks (8 × 8 pixels for JPEG/MJPEG and 4 × 4 pixels for AVC) are classified by their activity. The Standard Deviation (STD) is calculated for each pixel in a 3 × 3 window, then blocks are labeled depending on the maximum STD, as follows:

$$\text{Label} = \begin{cases} \text{StrongEdgeBlock} & \text{if Max STD} \geq 45 \\ \text{WeakEdgeBlock} & \text{if } 25 \geq \text{Max STD} < 45 \\ \text{StrongTextureBlock} & \text{if } 15 \geq \text{Max STD} < 25 \\ \text{WeakTextureBlock} & \text{if } 5 \geq \text{Max STD} < 15 \\ \text{SmoothBlock} & \text{if } 0 \geq \text{Max STD} < 5 \end{cases} \tag{2.10}$$

The spread parameter σ is then selected for each block following the

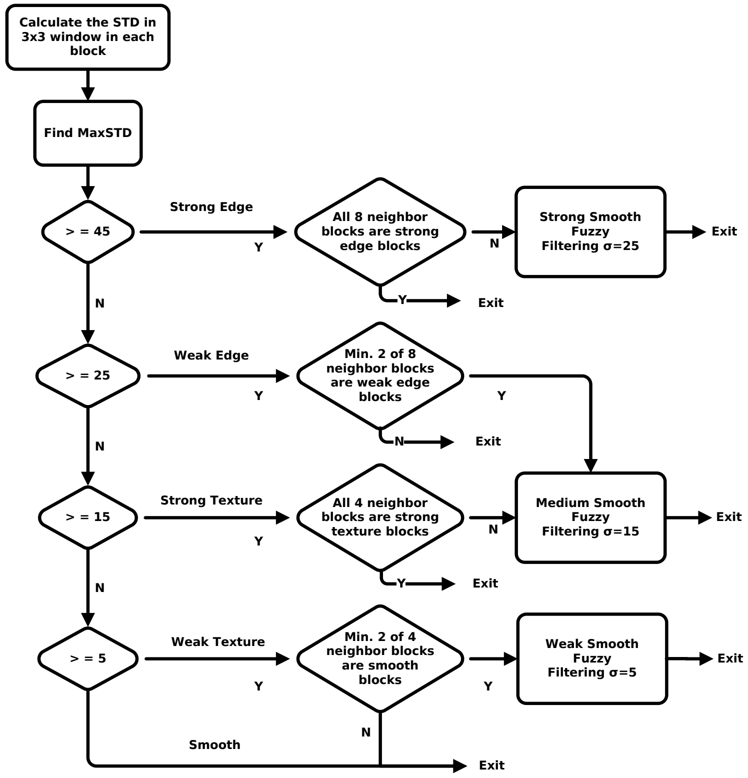

Calculate the STD in 3x3 window in each block
Find MaxSTD
> = 45
Strong Edge
Y
All 8 neighbor blocks are strong edge blocks
N
Strong Smooth Fuzzy Filtering σ=25
Exit
Y
Exit
N
> = 25
Weak Edge
Y
Min. 2 of 8 neighbor blocks are weak edge blocks
Y
N
Exit
N
> = 15
Strong Texture
Y
All 4 neighbor blocks are strong texture blocks
N
Medium Smooth Fuzzy Filtering σ=15
Exit
Y
Exit
N
> = 5
Weak Texture
Y
Min. 2 of 4 neighbor blocks are smooth blocks
Y
Weak Smooth Fuzzy Filtering σ=5
Exit
N
Smooth
Exit

Figure 2.3: Adaptive spread parameter (σ) selection for a block [2].

flow chart shown in Figure 2.3, depending on the labeling of neighboring blocks.

In the second step, fuzzy filtering is applied in the horizontal direction and, after anisotropic diffusion, on the vertical direction. Vertical artifacts are detected based on the values of the boundary pixels between blocks, shown in Figure 2.4. Blocks are 4×4 for H.264/AVC. For each pair of boundary pixels the absolute difference G_0 is found and compared with the absolute differences of the pixels left (L) and right (R) of the border. If $\text{MAX}(L_1, L_2, L_3) < G_0$ or $\text{MAX}(R_1, R_2, R_3) < G_0$, the row is marked as a boundary gap and is filtered by the 1D fuzzy filter. The same process is applied on horizontal artifacts. The filter is 5 pixels long and is applied on all pixels on the row or column within

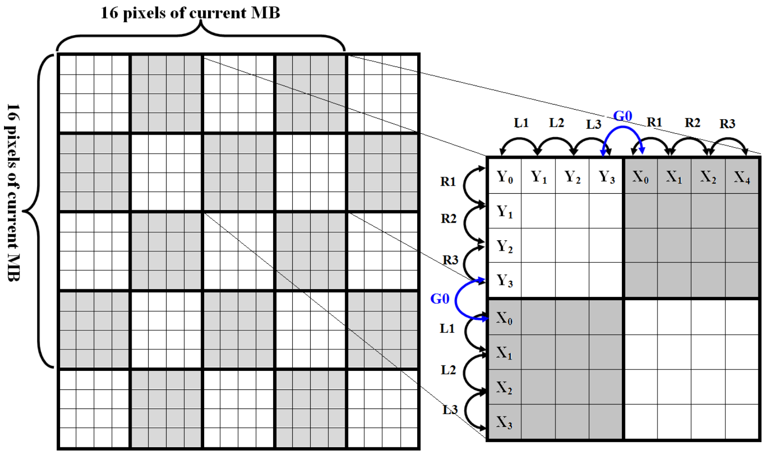


Figure 2.4: Detecting the vertical boundary gap in a row across the vertical block boundary for H.264/AVC video sequences [2].

the block. The values of the spread parameter σ is that selected in the previous step. For JPEG and MJPEG, the blocks are 8×8 and 4 pixel differences are considered to the left and to the right instead of just 3.

After the fuzzy filter has removed blocking from the image, anisotropic diffusion is applied. 1D anisotropic diffusion is chosen over the 2D version because the latter causes excessive blurring, as shown in Figure 2.5. After the k value is calculated from Eq. 2.4, the pixels are updated following

$$I_{i,j}^{n+1} = I_{i,j}^{n} + \frac{\Delta t}{2} d_{i,j}^{n}, \tag{2.11}$$

where $d_{i,j}^{n}$ is the approximated divergence derived from Eq. 2.1. In this work, $\frac{\Delta t}{2} \approx 0.125$. Note that Δt, i and j are discrete forms of t, x and j defined by $t = n\Delta t$ with $n = 0, 1, \ldots$, $x = ih$ with $i = 1, 2, \ldots, M-1$, and $y = jh$ with $j = 0, 1, \ldots, N-1$, respectively. It can be shown that Eq. 2.11 is equivalent to Eq. 2.2 in discrete form. Equation 2.11 is iterated until an ending condition is met or a number of iterations has been performed. In this work the number of iterations was set to 10.

Finally, after 1D fuzzy filter and 1D anisotropic diffusion have both been applied horizontally and vertically, 2D fuzzy filter with adaptive σ is applied. The values of σ are those determined in the first labeling step.

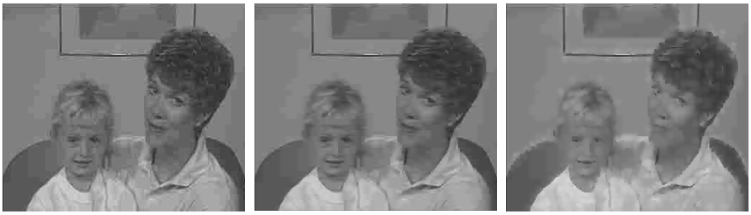


Figure 2.5: Comparison of 1D and 2D anisotropic diffusion equation applied to a JPEG compressed image for removing blocking artifact. All parameters are the same. JPEG compressed image with blocking effect (left), 1D anisotropic diffusion (center), 2D anisotropic diffusion (right) [2].

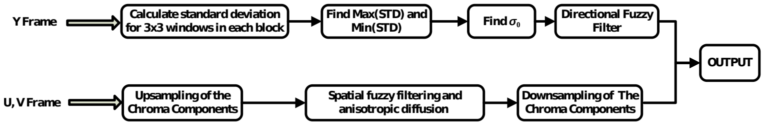

Y Frame
Calculate standard deviation for 3x3 windows in each block
Find Max(STD) and Min(STD)
Find σ_0
Directional Fuzzy Filter
OUTPUT
U, V Frame
Upsampling of the Chroma Components
Spatial fuzzy filtering and anisotropic diffusion
Downsampling of The Chroma Components

Figure 2.6: Flowchart of the method for video frames filtering [2].

2.3.4 Artifact Reduction in H.264/AVC Video Sequences

An alternative method for combating artifacts in video sequences compressed with H.264/AVC, still based on adaptive fuzzy filtering and anisotropic diffusion, is briefly described here. The flow chart of the method appears in Figure 2.6. YUV frames with downsampled chroma components are assumed.

In the Y component, the STD over a window of 3 × 3 pixels is calculated for all pixels, then the maximum and minimum STD are determined. A fuzzy filter with spread parameter determined by Eq. 2.9 is then applied. The amplitude parameter σ_A is calculated as

$$\sigma_A = \sigma_0 \left((1-\gamma) \times \frac{STD\,(I(i,j)) - STD_{min}}{STD_{max} - STD_{min}} + \gamma \right), \tag{2.12}$$

where $\mathrm{STD}_{\mathrm{min}}$ and $\mathrm{STD}_{\mathrm{max}}$ have just been defined and σ_0 is calculated as described in Figure 2.3.

Chroma components U and V are upsampled with bilinear interpolation to the same resolution of the Y frame, then the method using fuzzy filtering and anisotropic diffusion described in 2.3.3 (Figure 2.1) is applied. Finally, the components are downsampled back.

2.3.5 Experiments and Results

The methods for deblocking and deringing of images and video presented in the previous sections have been tested and compared against other algorithms, including those by Zhai et al. [27], Kim [28], Chen et al. [35], Liu et al. [36], Vo et al. [41], [44], [45], Tai et al. [47], Yao [39], Chebbo et al. [55] and Chen at al. [48]

2.3.5.1 Compressed Images

Six video sequences (Silent, Foreman, Mobile, Paris, News and Mother, shown in Figure B.6 in Appendix B) with Common Intermediate Format (CIF) resolution (352×288 pixels) and 50 frames have been compressed using MJPEG with scaling factor 4 for quantization.

The parameters of the proposed method were set as $\gamma = 0.5$ (Eq. 2.12), $\alpha = 0.5$ (Eq. 2.9), $\beta = 3.5$ (Eq. 2.9) and the set of neighboring pixels ω (Eq. 2.6) was set to 5×5 pixels.

The objective quality was measured in terms of Peak Signal to Noise Ratio (PSNR) and mean Structural Similarity (SSIM) [23] averaged over the three RGB channels. The average RGB-PSNR gain of the proposed method over all frames of the six sequences is 0.847, while the other methods go from 0.419 to 0.818. The average RGB-MSSIM gain of the proposed method over all frames of the six sequences is 0.038, while the other methods go from 0.014 to 0.029. For a visual comparison, the 6_{th} frames of Mother have been extracted and are shown in Figure 2.5. The proposed approach has the best visual quality, since it manages to reduce blocking and ringing at the same time, while other approaches are not equally able. More extensive results are available in [2].

2.3.5.2 Video Sequences

The algorithms have been applied also on MJPEG and H.264/AVC video sequences represented in the YUV color space. The proposed algorithm has been applied with the same parameters as in Section 2.3.5.1. Figure 2.8 shows the results of all algorithms on the 35_{th} frame of the Mobile sequence. The proposed method shows significantly reduce ringing artifacts and better color quality if compared to other methods. The visual comparison favour the proposed algorithm even more when the processed frames are played in sequence, since the quality is smoother.

Figures 2.9 and 2.10 compare YUV-PSNR and YUV-MSSIM values of all the tested methods for 45 frames of the Mobile sequence. The PSNR of the proposed method is consistently 0.55 dB higher that the compressed frames before post-processing and about 0.40 dB higher than the second best method. The respective MSSIM gains are about 0.05 and 0.02.

The method described in 2.3.4 has been tested on H.264/AVC compressed sequences. The six CIF sequences have been coded with the reference encoder using GOP lenght of 12 and GOP structure IBBPBB. . . , with a quantization parameter of 45. The in-loop filter was enabled. The proposed method averagely improved the PSNR of the Y component by 0.12 dB, that of the U component by 0.21 dB and that of the V component by 0.26 dB.

The proposed method for H.264/AVC video processing has also been compared with the other algorithms. This time the Foreman sequence have been encoded at 132 kbps bitrate (with variable quantization parameter). The proposed method has consistently higher PSNR and MSSIM compared, as shown in the plots of Figures 2.12 and 2.13. Visual inspection of the frames shown in Figure 2.11 shows that all the post-processing algorithms reduce the ringing and blocking artifacts compared to compressed frames. However, the proposed method maintains the sharpness of edges better, providing a more pleasant visual appearance. More extensive results are available in [2].

2.3.5.3 Subjective Evaluation

In order to verify the objective measurements, 16 test subjects were asked to rank video sequences played simultaneously. The Mobile sequence was compressed with MJPEG and post-processed with the proposed algorithm, 2D fuzzy filterning, the method by Kim [28], the one by Tai et al. [47] and the one by Zhai et al. [27]; similarly was done for the Foreman, but coding with H.264/AVC.

The results are summarized in Tables 2.1 and 2.2. In both cases the average ranking shows the best performance for the proposed method. With few 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 to estimate the rela-

Table 2.1: Subjective ranking of different methods (Mobile calendar compressed with MJPEG), compared with PSNR and MSSIM rankings and gains (Δ) relative to MJPEG coding [2].

Method	Mean subjective rank	Rank PSNR (Δ PSNR)	Rank MSSIM (Δ MSSIM)
MJPEG	5.563	6. (23.03 dB)	6. (0.786)
Zhai [27]	3.375	4. (+0.13 dB)	5. (+0.003)
Kim [28]	4.188	5. (+0.08 dB)	3. (+0.009)
2D fuzzy [45]	2.250	2. (+0.24 dB)	2. (+0.018)
Tai [47]	4.125	3. (+0.17 dB)	4. (+0.013)
Proposed	1.000	1. (+0.53 dB)	1. (+0.033)

Table 2.2: Subjective ranking of different methods (Foreman calendar compressed with H.264/AVC), compared with PSNR and MSSIM rankings and gains (Δ) relative to H.264/AVC coding [2].

Method	Mean subjective rank	Rank PSNR (Δ PSNR)	Rank MSSIM (Δ MSSIM)
H.264/AVC	4.750	6. (32.24 dB)	6. (0.8684)
Zhai [27]	1.875	2. (+0.11 dB)	5. (+0.0000)
Kim [28]	4.688	5. (+0.03 dB)	3. (+0.0022)
2D fuzzy [45]	3.813	4. (+0.05 dB)	4. (+0.0004)
Tai [47]	4.188	3. (+0.07 dB)	2. (+0.0017)
Proposed	1.250	1. (+0.17 dB)	1. (+0.0029)

tive performance of artifact removal algorithms. For MJPEG, all post-processing methods are visually ranked better than direct decoding without post-processing. The proposed method is preferred to all other algorithms. More extensive results are available in [2].

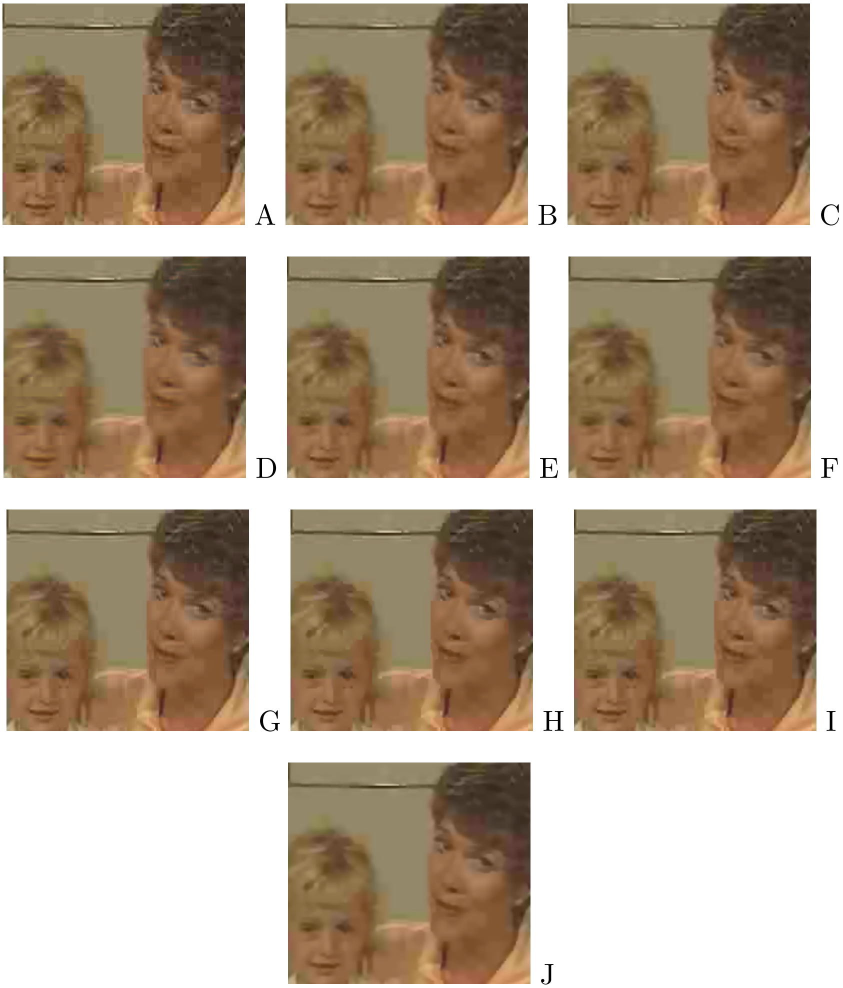


Figure 2.7: Comparison of filtered results on zoomed JPEG images [2]: (A) the compressed image-4Q (PSNR=31.02, MSSIM=0.839); (B) Chen [35] (PSNR=31.83, MSSIM=0.869); (C) Liu [36] (PSNR=31.61, MSSIM=0.862); (D) Kim [28] (PSNR=31.69, MSSIM=0.861); (E) Chebbo [55] (PSNR=31.65, MSSIM=0.864); (F) 2D fuzzy filter (PSNR=31.77, MSSIM=0.865); (G) Zhai [27] (PSNR=31.64, MSSIM=0.862); (H) Tai [47] (PSNR=31.84, MSSIM=0.869); (I) Yao [39] (PSNR=31.78, MSSIM=0.865) and (J) the proposed method (PSNR=32.10, MSSIM=0.872).

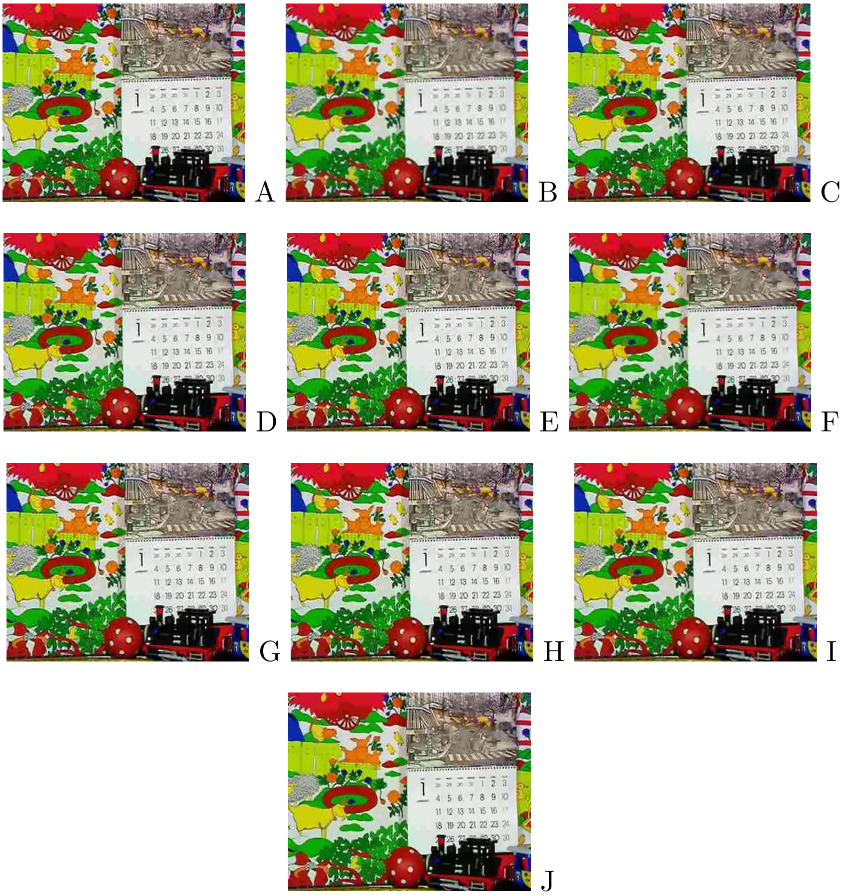


Figure 2.8: Comparison of filter results for MJPEG sequences (35_{th} frame of Mobile) [2]. (A) Compressed (PSNR=23.03, MSSIM=0.786); (B) Chen [35] (PSNR=22.58,MSSIM=0.774); (C) Liu [36] (PSNR=22.88, MSSIM=0.782); (D) Kim [28] (PSNR=23.28, MSSIM=0.794); (E) Chebbo [55] (PSNR=23.13,MSSIM=0.793); (F) 2D fuzzy filter (PSNR=23.26, MSSIM=0.804); (G) Zhai [27](PSNR=23.23, MSSIM=0.794); (H) Tai [47] (PSNR=23.42, MSSIM=0.798); (I) Yao [39] (PSNR=23.49, MSSIM=0.784) and (J) the proposed method (PSNR=23.56, MSSIM=0.819).

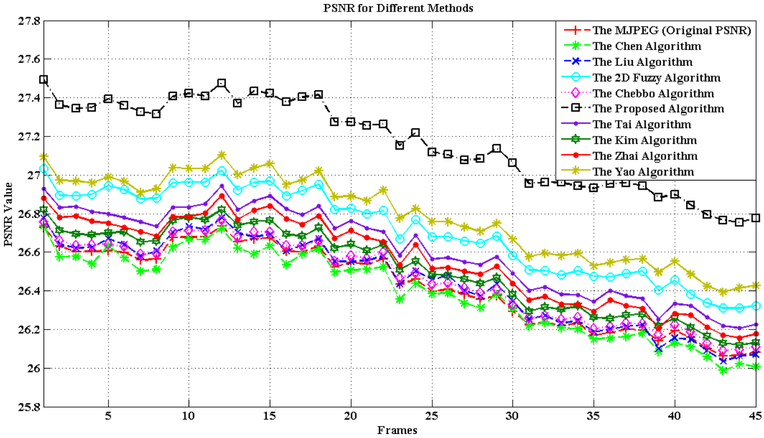


Figure 2.9: Comparison on PSNR of simulated methods for mobile sequence [2] (average PSNR: MJPEG=26.43, Chen=26.39, Liu=26.45, Kim=26.51, Chebbo=26.46, Zhai=26.56, 2D fuzzy=26.71, Tai=26.61, Yao=26.78, Proposed=27.17).

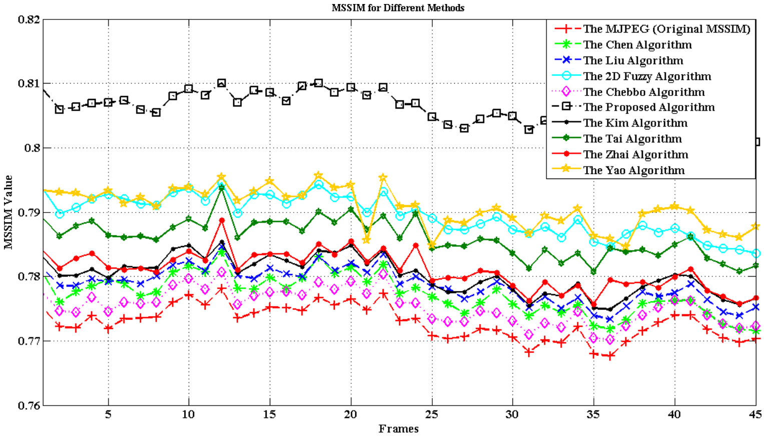


Figure 2.10: Comparison on MSSIM of simulated methods for Mobile sequence [2] (average MSSIM: MJPEG=0.773, Chen=0.777, Liu=0.779, Kim=0.780, Chebbo= 0.775, Tai= 0.786, Zhai=0781, 2D fuzzy=0.789, Yao=0.791, Proposed =0.806).

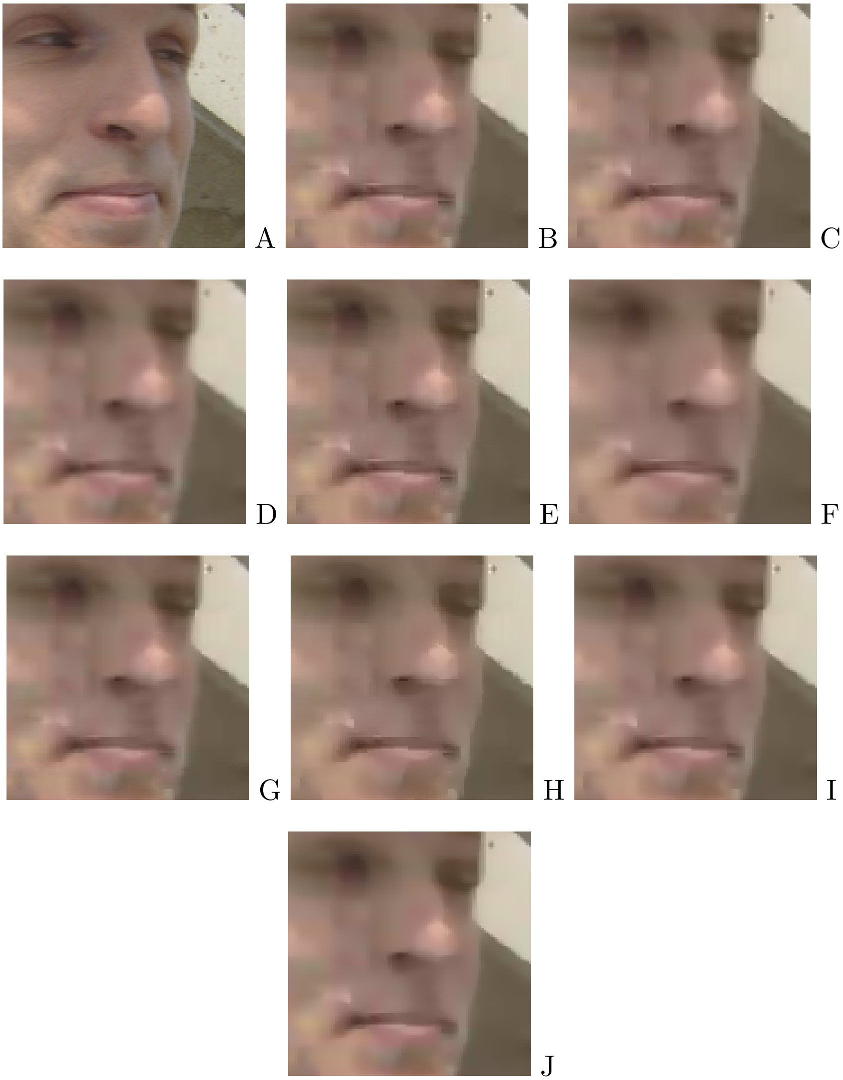


Figure 2.11: Comparison of filter results for H.264/AVC sequences (3_{rd} frame of Foreman zoomed) [2]. (A) original frame; (B) compressed with in-loop filtering (PSNR=32.30, MSSIM=0.869); (C) Kim [28] (PSNR=32.32, MSSIM=0.872); (D) Chebbo [55] (PSNR=32.32, MSSIM=0.870), (E) 5/3 Haar MCTF [48] (PSNR=32.11, MSSIM=0.873), (F) Zhai [27] (PSNR=32.45, MSSIM=0.873); (G) 2D fuzzy filter (PSNR=32.13, MSSIM=0.872), (H) Tai [47] (PSNR=32.44, MSSIM=0.874); (I) Yao [39] (PSNR=32.42, MSSIM=0.873) and (J) the proposed method (PSNR=32.57, MSSIM=0.876).

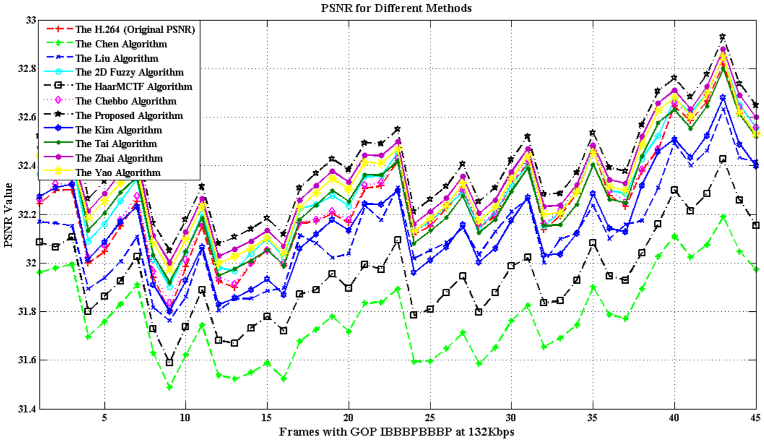


Figure 2.12: Comparison on PSNR of simulated methods for H.264/AVC sequence (Foreman sequence) [2].

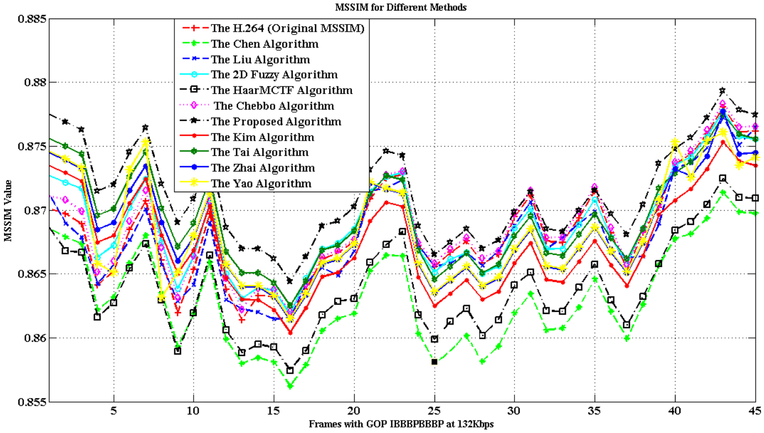


Figure 2.13: Comparison on MSSIM of simulated methods for H.264/AVC sequence (Foreman sequence) [2].

2.4 Adaptive Deblocking and Deringing of H.264/AVC video

This section presents a method to reduce blocking and ringing artifacts in H.264/AVC coded video. Deblocking is performed 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. This method was presented in [3].

2.4.1 Deblocking Step

The deblocking step goes through 3 phases, illustrated in Figure 2.14: measuring of the frame quality, selection of decision modes and two-step deblocking filtering. The frame quality is used to select the decisions mode, which determine the local strength of the filters. This way, flat area can be filtered more aggressively and detailed areas more weakly.

Depending on the frame type (I, P or B) and on the quantization steps, pixels are distorted with different degrees and quality. The quality measure Q_M is defined to approximate MSE for each pixel in I, P and B frames:

$$Q_M = \sqrt{12 \times \text{MSE}} \tag{2.13}$$

The lower Q_M, the higher the quality. The curves shown in Figure 2.15 were obtained by measuring the Mean Squared Error (MSE) of the luminance components of H.264/AVC decoded sequences. All settings and testing in later experiments for this method are based on these curves. QP determines the quantizer step size Q_s [56]. The Laplacian distribution is used to model the MSE as shown in Figure 2.15. The results indicate that I frames provide the lowest error, and that P frame have lower error than B frames at the same Q_s. As QP increases, degradations increases in all frames, while the quality differences between I, P and B frames decreases. The quantization step size Q_s can be determined from the QP and the frame type, then the MSE can be estimated from the curves in Figure 2.15 and finally Q_M can be calculated using Eq.2.13. The decision modes and segmentation step use the following

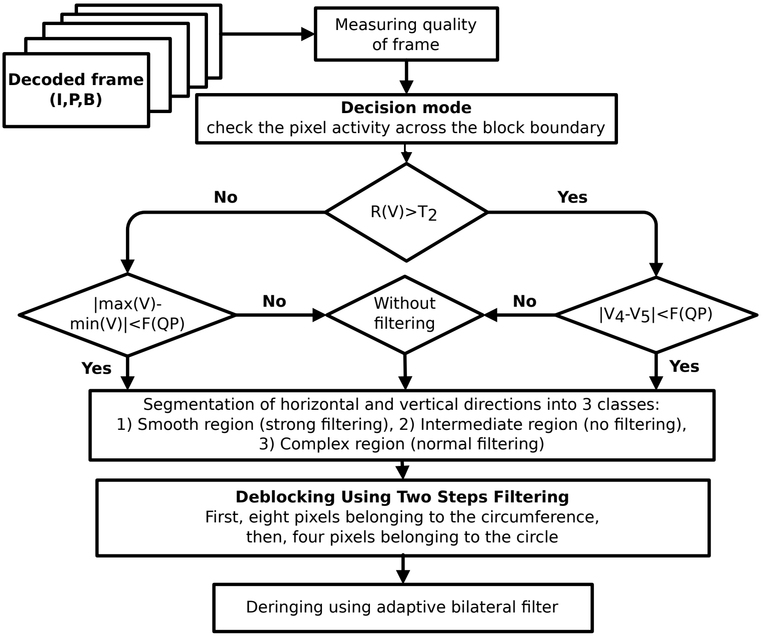

Decoded frame
(I,P,B)
Measuring quality of frame
Decision mode
check the pixel activity across the block boundary
R(V)>T2
No
Yes
\|max(V)-min(V)\|<F(QP)
No
Without filtering
No
\|V4-V5\|<F(QP)
Yes
Yes
Segmentation of horizontal and vertical directions into 3 classes:
1) Smooth region (strong filtering), 2) Intermediate region (no filtering),
3) Complex region (normal filtering)
Deblocking Using Two Steps Filtering
First, eight pixels belonging to the circumference,
then, four pixels belonging to the circle
Deringing using adaptive bilateral filter

Figure 2.14: Flowchart of the proposed debloking and deringing algorithm based on decision modes and bilateral filtering [3].

function of Q_M:

$$F(QP) = \sqrt{Q_M} \tag{2.14}$$

In the second phase, areas to be filtered are classified into flat and complex regions based on the pixel activity of horizontal and vertical pixel vectors at each 4×4 block boundary, as shown in Figure 2.16. Let V and v_i denote the eight-pixels vector and pixel values, respectively. The activity is given by:

$$R(V) = \sum_{i=1}^{7} \phi(v_i - v_{i+1}), \tag{2.15}$$

where $0 \leq R(V) \leq 7$ and

$$\phi(\Delta) \begin{cases} 1, & \text{if } |\Delta| < T_1 \\ 0, & \text{otherwise} \end{cases}, \tag{2.16}$$

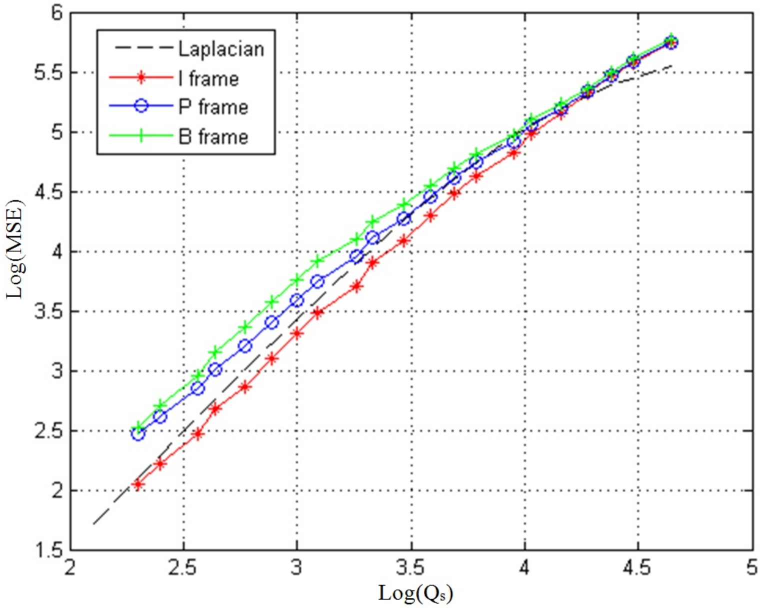
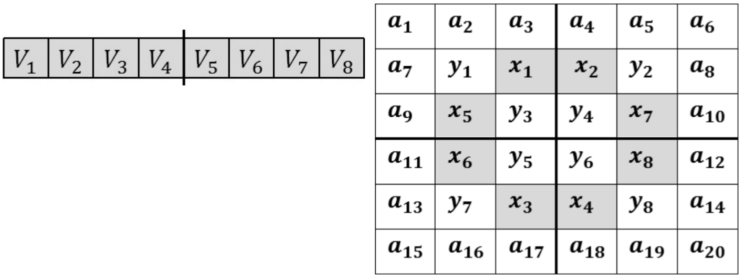


Figure 2.15: MSE vs Q_s measured on the Mobile sequence (CIF); rate control is disabled, different QP values chosen for the different points [56].

where T_1 is a fixed threshold (should be set to a small value). The activity factor reflects the activity in V across block boundaries and the number of detected edges within it. 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 $F(QP)$, V is assumed to be in a complex region and a filter for complex regions is applied. If the value of $R(V)$ is bigger than $F(QP)$, then no filtering is needed. If $R(V) > T_2$, the absolute difference between the two pixel values on the sides of the block boundary (v_4 and v_5) is considered: if it is smaller than $F(QP)$, V is assumed to be in a smooth region and is processed with a weak filter, otherwise it does not need filtering. Here, $T_1 = 6$ and $T_2 = 2$. Based on the decisions modem, the frame is partitioned in three regions in the horizontal and vertical directions,: no filtering (N), weak filtering (W) and hard filtering (H). Figure 2.17 shows an example of partitioning.

Two steps filtering is done after each pixel has been labeled horizontally and vertically. A 6 × 6 window is centered at the intersection



a_1	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_3	y_4	x_7	a_{10}
a_{11}	x_6	y_5	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}

Figure 2.16: 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 (Left). Position of pixels in the filtering window (6 × 6) for the two-step filtering (Right) [3].

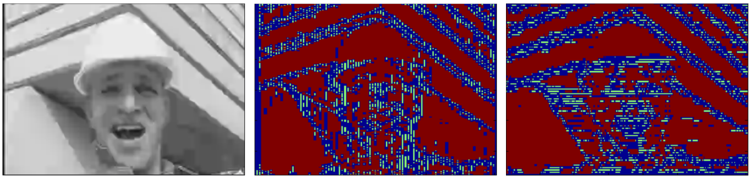


Figure 2.17: Decision mode of the 25_{th} frame of the Foreman sequences: coded frame (left), horizontal direction modes (center), vertical direction modes (right). The red areas are categorized as smooth and are applied a hard (H) filter, the blue areas are categorized as intermediate and no filtering (N) is applied, the green areas are categorized as complex and a weak (W) filter is applied [3].

of four 4 × 4 pixel blocks as shown in Figure 2.16. This is done for all intersections. In the first step, only eight pixels are filtered ($x_1 \ldots x_8$ in Figure 2.16). If filtering is selected in only one direction (horizontal or vertical), 1D filters are required. For instance, in NW or WN modes only a weak 1D filter is applied on the pixel on the vertical or horizontal directions, respectively. When both directions are labeled for weak or hard filtering, 2D filtering is applied on the pixel. Equation 2.17 shows the updated values of pixel x_{1u} in different modes ($\gg$ is the bitshift operator):

$$x_{1u} = \begin{cases} y_1 + 5x_1 + 3x_2 - y_2 \gg 3 & \text{if NW} \\ 2a_7 + y_1 + 2x_1 + x_2 + y_2 + a_8 \gg 3 & \text{if NH} \\ 4(x_1 + a_3) + 2(x_2 + y_2 + y_3) + y_1 + y_2 \gg 4 & \text{if WH} \\ \begin{array}{c} 4(a_2 + a_3 + a_4 + x_1) + 2(y_1 + x_5 + y_4 + y_6) + \\ a_7 + x_2 + y_2 + a_8 + y_3 + y_5 + x_3 + x_6 \gg 5 \end{array} & \text{if HH} \end{cases} . \tag{2.17}$$

The other pixels are filtered similarly. In the second filtering step, pixels $y_3 \dots y_6$ are filtereed according to the decision mode based partitioning. The filtered values of $x_1 \dots x_8$ are included in the calculation. Pixel y_{3u} is updated as follows:

$$y_{3u} = \begin{cases} 3x_5 + y_3 + y_7 - y_4 \gg 2 & \text{if NH} \\ 2x_5 + 5y_3 + 3y_4 - 2x_7 \gg 3 & \text{if NW} \\ 6x_5 + 4y_3 + 2y_4 + 4x_7 + 2y_5 - x_1 - x_3 \gg 4 & \text{if HW} \\ x_5 - x_7 + x_1 - x_3 + 4y_3 + 2y_5 + 2y_4 \gg 3 & \text{if WW} \\ 2(x_5 + x_7 + x_1 + x_3) + y_5 + 6y_3 + y_4 \gg 4 & \text{if HH} \end{cases} . \tag{2.18}$$

The remaining pixels are calculated similarly.

2.4.2 Bilateral Filter for Deringing

An adaptive bilateral filter is used to remove ringing artifacts after blocking artifacts have been removed. The bilateral filter is a nonlinear weighted averaging filter, obtained by combining two Gaussian filters, one working in the spatial domain, the other working in the intensity domain [57]. The bilateral filter can preserve edges while doing spatial smoothing. The output of the filter at pixel location x can be formulated as follows:

$$J(x) = \frac{1}{Z} \sum_{y \in \psi(x)} e^{\frac{-\|(y-x)\|^2}{2\sigma_d^2}} e^{\frac{-|(I(y)-I(x))|^2}{2\sigma_r^2}} I(n), \tag{2.19}$$

where σ_d and σ_r are parameters controlling the fall-off of the weights in the spatial and intensity domains, respectively. $\psi(x)$ is the spatial neighborhood of pixel $I(x)$ and Z is a normalization constant.

For deringing, σ_d and σ_r should be chosen carefully for every block to avoid oversmoothing texture regions and to preserve edge regions. For a smooth region, the value of the σ_d can be large, otherwise it should be small. Here, each 4×4 block is classified into one of the four categories (strong edge, weak edge, texture, smooth) depending on the maximum STD measured on a 4×4 window around each pixel in the block. The value if σ_d is set as follows:

$$\sigma_d = \begin{cases} \text{StrongEdge}(\sigma_d = 0.8) & \text{if MaxSTD} \in [35, \infty) \\ \text{WeakEdge}(\sigma_d = 1.8) & \text{if MaxSTD} \in [25, 35) \\ \text{Texture}(\sigma_d = 2.8) & \text{if MaxSTD} \in [15, 25) \\ \text{Smooth}(\sigma_d = 3.8) & \text{if MaxSTD} \in [0, 15) \end{cases} \tag{2.20}$$

The optimal σ_r value of is $\sigma_r = \alpha \times \sigma_n$, where σ_n is the STD of the image noise, which can be estimated [56]. Here, the value of α is set to $\frac{1}{3}$ in each 4×4 block.

2.4.3 Experiments and Results

The proposed algorithm was evaluated on H.264/AVC video sequences by comparing it with other spatial post-processing algorithms [1], [27], [28], [39], [47], [55], [58]–[60]. The GOP structure was set to $(\text{IPPB})_{12}$. Two different types of experiments have been performed. In the first experiment, the algorithm was applied with two different QP (35 and 45) with the in-loop deblocking filter enabled. In the second experiment the in-loop deblocking filter was disabled. The algorithms were applied on the first 100 frames of several CIF (4:2:0) sequences: Akiyo, Bus, Coastgard, Container, Cycling, Foreman, Hall, Mobile, and Mother, shown in Figure B.6 in Appendix B. The performance of the algorithms has been compared in terms of Weighted-PSNR (W-PSNR) and Weighted-MSSIM (W-MSSIM), where the weights for the luma and chroma components are $\frac{2}{3}$ and $\frac{1}{6}$, respectively [2], [61].

Table 2.3 summarizes the results of the first experiment. The proposed approach reaches higher PSNR and MSSIM compared to the other algorithms.

Table 2.4 shows the PSNR results for all the test video sequences (Figure B.6). As shown in this table, the proposed algorithm has better performance compared to the other selected algorithms [27], [59].

Table 2.5 shows the results of the second experiment, where the in-loop filter is disabled on the Akiyo video sequences. The proposed

Table 2.3: The average results of post-processing H.264/AVC video test sequences using different algorithms [3].

Metric	W-PSNR		W-MSSIM	
QP	35	45	35	45
H.264/AVC	34.76	30.51	0.906	0.810
[1]	35.09	30.63	0.911	0.812
[28]	34.09	30.35	0.898	0.806
[55]	35.03	30.52	0.910	0.811
[47]	34.66	30.49	0.901	0.809
[58]	34.82	30.57	0.907	0.813
[27]	35.03	30.59	0.910	0.813
[39]	35.00	30.36	0.907	0.809
[59]	35.04	30.39	0.909	0.810
[60]	34.82	30.44	0.908	0.809
Proposed	35.18	30.62	0.911	0.814

Table 2.4: The average W-PSNR results for all the test video sequences using the selected algorithms.

Algorithm	H.264/AVC		Zhang et al.[59]		Zhai et al.[27]		Proposed	
QP	35	45	35	45	35	45	35	45
Akiyo	37.65	32,70	37.75	32.87	37.71	32.83	37.83	32.86
Bus	33.26	28.36	33.18	28.43	33.22	28.41	33.36	28.45
Coastguard	34.43	29.93	34.31	30.02	34.41	30.01	34.54	30.01
Container	34.91	30.07	34.91	30.17	34.93	30.13	37.98	30.21
Cycling	35.11	30.34	35.09	30.42	35.11	30.41	35.18	30.41
Foreman	33.19	31.19	35.66	31.28	35.56	33.25	35.81	33.27
Hall	35.72	30.97	35.83	31.14	38.82	31.06	35.95	31.13
Mobile	30.88	25.79	30.92	25.81	30.82	25.79	30.85	25.84
Mother	37.64	33.18	37.69	33.37	37.71	33.37	37.81	33.38

algorithm reaches higher PSNR and MSSIM when compared to the in-loop filter alone. Figure 2.18 visually compares the in-loop filter and proposed algorithm. Blocking and ringing artifacts are more effectively attenuated in both images, resulting in a better perceptual quality for the reconstructed video.

Table 2.5: Results of H.264/AVC video when the in-loop filtering is disabled/enabled and with the proposed algorithm on the Akiyo sequence [3].

Metric	W-PSNR		W-MSSIM	
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

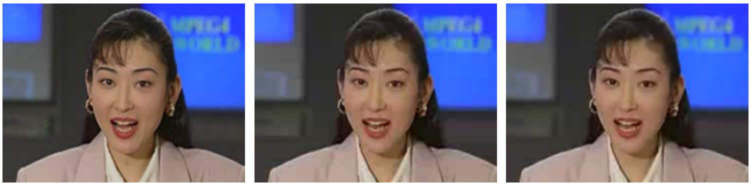


Figure 2.18: The comparison of filter result on the 75_{th} frame of Akiyo: Compressed frame (left), In-loop filter (center), Proposed algorithm (right) [3].

2.5 Post-Processing of Video Sequences Using Deblocking and Adaptive Trilateral Filtering

Previously, some algorithms were proposed to extend the bilateral filter [62]–[64]. The trilateral filter presented in [62] combines two modified bilateral filters with a novel image-stack scheme for fast region-finding. It is used to reduce the impulse noise in images. They also demonstrated that their filter can be used for reducing the contrast and making a clear image. In [63], Wong et al. proposed a new extension of the bilateral filter for bio-medical images, that can achieve edge-preserving smoothing with a narrow spatial window in a few iterations. The algorithm was also capable to reduce the coding artifact of still images [64]. To define and add a new term of the bilateral filter, a dual edge detection method was used to extract the texture and edge regions. In addition, the spread parameters of their filter was selected manually.

This section presents another method to reduce blocking and ringing artifacts in compressed video signal. For deblocking, the proposed method uses the deblocking step as mentioned in [3]. An adaptive derining method is proposed through a trilateral filter. To make trilateral filter, a new multiplicative filter is added to the spatial and intensity filter of bilateral filter. The spread parameters of trilateral filter are selected adaptively using texture and edge mapping. The analysis of objective and subjective experimental results shows that the proposed algorithm is successful in artifact reduction of low bit-rate H.264 video sequences.

2.5.1 Trilateral Filter for Removing Ringing Artifact

As mentioned before, the behavior of the bilateral filter depends on the spatial and intensity spread parameters (σ_s, σ_r). The ringing artifact is a kind of edge related distortion and arises the near edges in highly compressed frames. In order to improve an image quality and preserve its edge details, edge and texture detection are often employed in deringing methods. To have better deringing performance, the spread parameters should be tuned. Therefore, in some cases, it is better to increase the range of σ_s and σ_r. However, if some image details are distributed within the texture area, the range of σ_s and σ_r should be reduced to avoid over smoothing of the image details. Therefore, to adjust the σ_s and σ_r according to the local features of an image an extra Gaussian multiplicative filter is added.

In order to design the multiplicative filter, the texture region should be detected. For this, the deviation of each pixel and its neighborhood is considered. First, the standard deviation (STD) is calculated in a 3×3 window around each pixel. Thereby, a matrix of standard deviations is obtained ($STD\ map$). Then, the maximum STD ($MaxSTD$) within 2×2 blocks of the $STD\ map$ are compared with two predetermined threshold values. Any block region exceeding the threshold (T_3 and T_4) is a texture region in the original image. The texture detection algorithm is described in Algorithm. 1. In this work the T_3 and T_4 were set to 15, 25. An example of the texture map is shown in Figure 2.19.

After detecting the texture regions, the Gaussian multiplicative filter which is a texture activity filter (M) is defined by:

$$M_{(m,n)} = \begin{cases} e^{\frac{-\|n-m\|^2}{2\sigma_t^2}}, & \text{if } \textit{texture} = I \\ 1, & \text{if } \textit{texture} = 0 \end{cases}, \tag{2.21}$$

Algorithm 1 The texture detection algorithm.

```
Step 1: Calculate standard deviation ($STD$) for each pixel of the input image ($I$) in a $3 \times 3$ window and form a $STD$ map with the same size as the input image.
Step 2: Convert the $STD$ map into $2 \times 2$ blocks and find $MaxSTD$ in each block
Step 3: For all blocks ($k = 1...N$) ($N$ is the number of blocks)
if $T_3 < MaxSTD_k < T_4$ then
    $texture_k = I_k$
else
    $texture_k = 0$
end if
Step 4: Extract texture image with the same size of the input image.
```

where σ_t is the texture spread parameter that is a scaled version of the spatial spread parameter (σ_s).

Similar to the bilateral filter defined in 2.19, the trilateral filter is introduced as:

$$T(m) = \frac{1}{P(m)} \sum_{n \in \psi(m)} S(m,n)M(m,n)R(I(m), I(n))I(n) \qquad (2.22)$$

where S and R are the spatial and intensity Gaussian kernels, respectively, M is the third Gaussian kernel of the trilateral filter, $P(m)$ is a normalization constant and $\psi(m)$ is the spatial neighborhood of pixel m as mentioned in Eq. 2.19.

The texture spread parameter (σ_m) is calculated as follows:

$$\sigma_m = \sigma_s / \sqrt{0.2 \times texture}, \qquad (2.23)$$

where the *texture* is the texture image map formed in Algorithm. 1. When there are no textures within an area, *texture* approaches toward zero. Thus, σ_t tends to be infinite, and the activity filter in Eq. 2.21 tends to be flat (one), which means that the trilateral filter in Eq. 2.22 tends to bilateral. On the contrary, in the textural area, with the *texture* value increasing, the range of activity texture filter shrinks, which equivalently attenuates the trilateral filter.

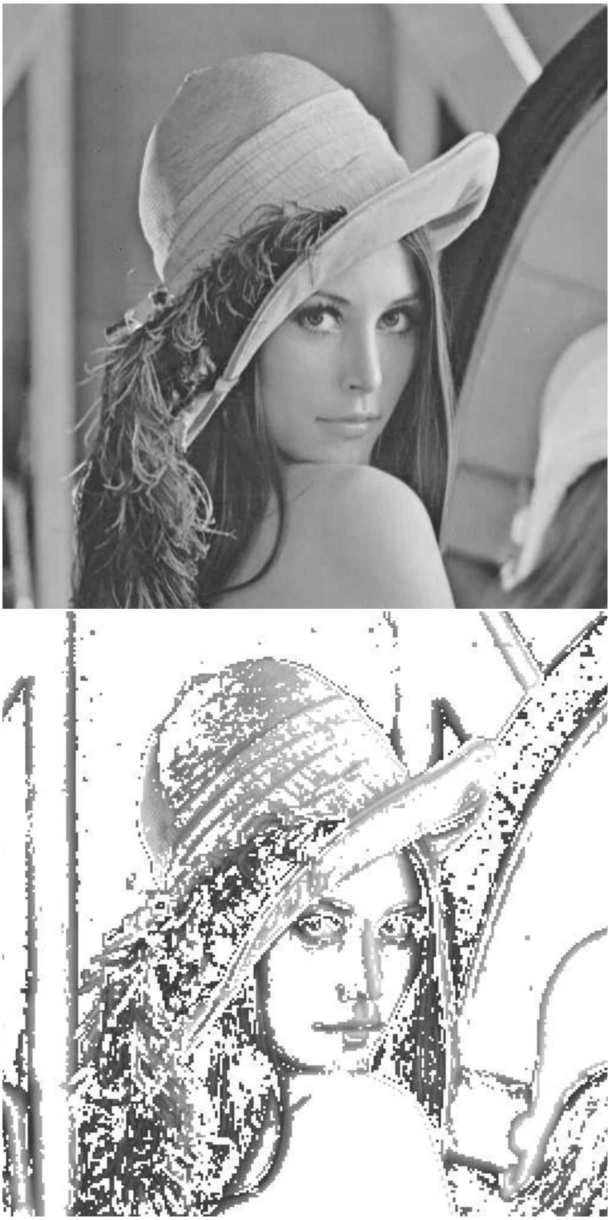


Figure 2.19: Texture map of the *Lena* image: Original image (left), Texture regions (right).

The optimal value for σ_s and σ_r in the trilateral filter is calculated as explained in Section 2.4.2.

2.5.2 Experiments and 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], [27], [28], [39], [45], [47], [55], [58]–[60]. The bilateral method was presented in [3] and the trilateral method is proposed in this section. The GOP structure was defined as $(IPPB)_{12}$. Two different types of experiments were 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 quality improvement obtained by applying the proposed algorithm to the *Foreman* sequence compressed with different bit rates. Nine CIF (4:2:0) test sequences were chosen: *Akiyo*, *Bus*, *Coastguard*, *Container*, *Cycling*, *Foreman*, *Hall*, *Mobcal*, *Mother and Daughter*. These sequences shown in Figure B.6 in Appendix B. The algorithms were applied on the first 100 frames of each sequence. The quality of different algorithms was compared in terms of Y-PSNR, W-PSNR, Y-MSSIM and Y-SSD. Y denotes the luma component and W the weighting of components, where the luma and chroma components have a weight of 2/3 and 1/6, respectively [3].

Figures 2.20 and 2.21 compare the Y-PSNR, and Y-MSSIM values of the tested algorithms for *Foreman* sequence. The plots clearly demonstrate that the proposed method achieves a consistent average PSNR gain of about 0.317 dB and the average MSSIM gain about %3.4, which are better than the other methods.

Figure 2.22 compares the enhanced frame obtained using different algorithms, including the proposed algorithm applied on the 56^{th} frame of the *Foreman* video sequence. The enhanced frame obtained by the proposed method (Figure 2.22F) shows significantly reduced artifacts and better quality than the other algorithms, including the MCSTF [45]. 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 consistant quality with significantly reduced artifacts.

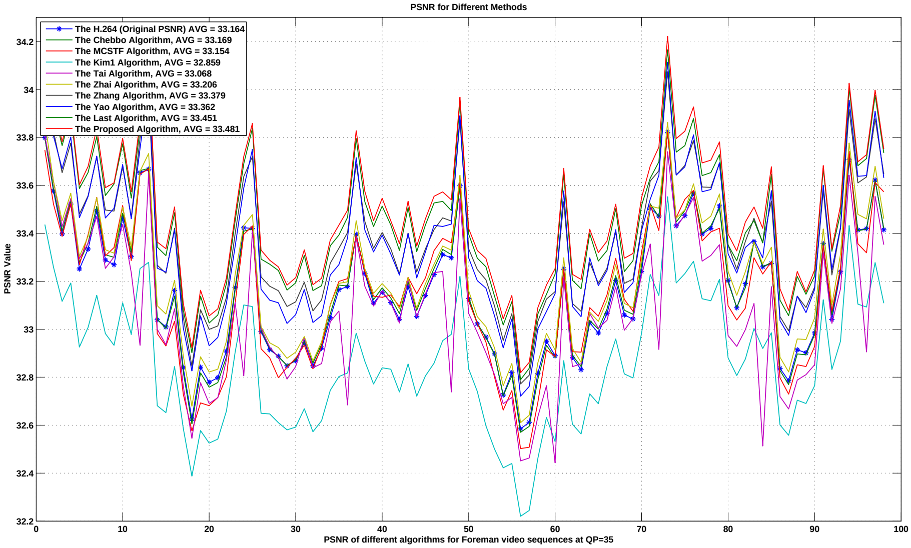


Figure 2.20: Comparison on PSNR of simulated methods for Foreman sequence with QP=35.

The comparative objective results for the first experiment are summarized in Table 2.6. It can be seen that the proposed algorithm achieves higher PSNR and MSSIM compared to the other methods. The overall average Y-PSNR, W-PSNR improvement of the proposed algorithm are for QP $= 35$: 0.13 and 0.12 and for QP $= 45$: 0.45 and 0.15, respectively. These results demonstrate the effectiveness of the proposed algorithm.

The selected results for all the test video sequences are presented in Table 2.7. It can be seen that the performance of the proposed algorithm is higher than the other algorithms for both QP 35 and 45.

To evaluate the effectiveness of the proposed algorithm in the temporal dimension, the Sum of Squared Differences (SSD) is also computed for all test sequences for both quantization parameters (35 and 45). High SSD values show more flickering. The average results are presented in Table 2.8. These results show that some algorithms have high SSD which indicates more flicker compared to the original compressed sequence. Whereas, the others have low SSD which shows a reduction in flickering. The proposed algorithm shows the lowest level of flickering artifact measured by SSD. The selected results for all videos are

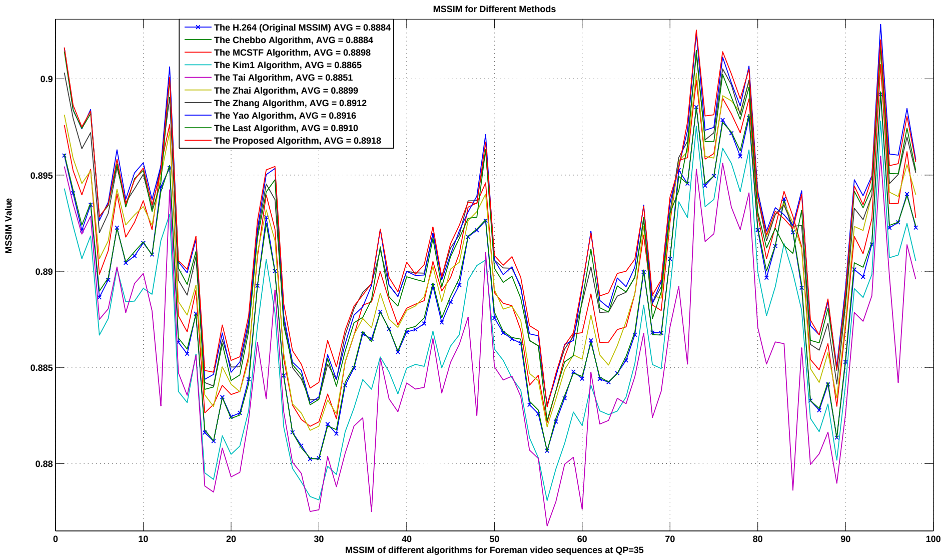


Figure 2.21: Comparison on MSSIM of simulated methods for Foreman sequence with QP=35.

shown in Table 2.9. Compared with MCSTF [45] and Bilateral [3], the proposed algorithm has less flickering artifact.

To study the performance of the proposed algorithm more comprehensively, the *Foreman* sequence was compressed using different bit rates. It was evaluated by the PSNR and SSD metrics. As shown in Fig. 2.23, the proposed algorithm improved the spatial and temporal quality simultaneously, compared with H.264/AVC video sequences. As mentioned before, the GOP structure in our work was $(IBBP)_{12}$ while in MCSTF (spatio-temporal) algorithm [45], it was *IBBBPBBBP* and their improvement was around +0.2 dB Y-PSNR. However, the proposed algorithm does not require the motion compensation stage and the spatio-temporal filter used in MCSTF algorithm. Therefore, the computational complexity of the proposed method is lower compared to the MCSTF.

To compare the visual qualities, the results obtained by the proposed and bilateral [3] methods are shown in Fig. 2.24 for a compressed JPEG image. As can be seen, the proposed trilateral method has better performance for removing artifact and improving the image quality than

Table 2.6: The average results of post-processing H.264/AVC video test sequences using different algorithms (The results for [45] was only provided for the luma component).

Metric	Y-PSNR		W-PSNR		Y-MSSIM	
QP	35	45	35	45	35	45
H.264/AVC	32.67	26.85	34.76	30.51	0.8905	0.7402
[2]	32.76	26.94	35.09	30.63	0.8907	0.7439
[28]	31.36	26.63	34.09	30.35	0.8729	0.7329
[55]	32.47	26.86	35.03	30.52	0.8904	0.7409
[47]	32.31	26.83	34.66	30.49	0.8792	0.7385
[39]	32.68	26.93	35.00	30.36	0.8871	0.7428
[27]	32.69	26.88	35.03	30.59	0.8905	0.7423
[59]	32.73	26.93	35.04	30.39	0.8893	0.7436
[60]	32.44	26.81	34.82	30.44	0.8879	0.7393
[45]	32.50	26.84	-	-	0.8883	0.7401
[65]	29.75	26.02	32.85	29.88	0.8519	0.7157
Bilateral [3]	32.78	26.95	35.18	30.62	0.8906	0.7436
Trilateral	32.80	26.97	35.21	30.66	0.8916	0.7538

the previous method.

Table 2.7: The average Y-PSNR results for all the test video sequences using the selected algorithms (The sequences are 1:*Akiyo*, 2:*Akiyo*, 3:*Coastguard*, 4:*Container*, 5:*Cycling*, 6:*Foreman* 7:*Hall*, 8:*Mobile* and 9:*Mother*, respectively).

X	H.264/AVC		Zhang et al.[59]		Bilateral [3]		Trilateral	
QP	35	45	35	45	35	45	35	45
1	36.25	30.34	36.31	30.44	36.33	30.49	36.37	30.52
2	30.36	24.19	30.35	24.24	30.39	24.25	30.41	24.26
3	30.55	25.34	30.42	25.36	30.55	25.34	30.56	25.36
4	32.68	26.82	32.66	26.92	32.74	26.96	32.74	26.96
5	32.61	26.94	32.68	26.99	32.70	26.99	32.71	27.01
6	33.16	27.99	33.38	28.11	33.45	28.18	33.48	28.21
7	34.09	27.95	34.23	28.09	34.27	28.19	34.29	28.22
8	28.77	22.84	28.94	22.91	28.95	22.93	28.97	22.95
9	35.58	30.38	35.59	30.46	35.62	30.47	35.63	30.50

Table 2.8: The average results of the luma component of Sum of Squared Differences Y-SSD metric on H.264/AVC video using different algorithms.

Metric	Y-SSD	
QP	35	45
H.264/AVC	48.79	144.44
[2]	47.73	142.11
[28]	60.41	144.74
[55]	48.75	144.21
[47]	52.15	145.73
[39]	48.01	142.26
[27]	48.59	173.81
[59]	47.67	142.61
[60]	50.21	145.29
[45]	51.47	142.12
[65]	71.44	157.49
Bilateral [3]	47.69	141.77
Trilateral	47.37	141.45

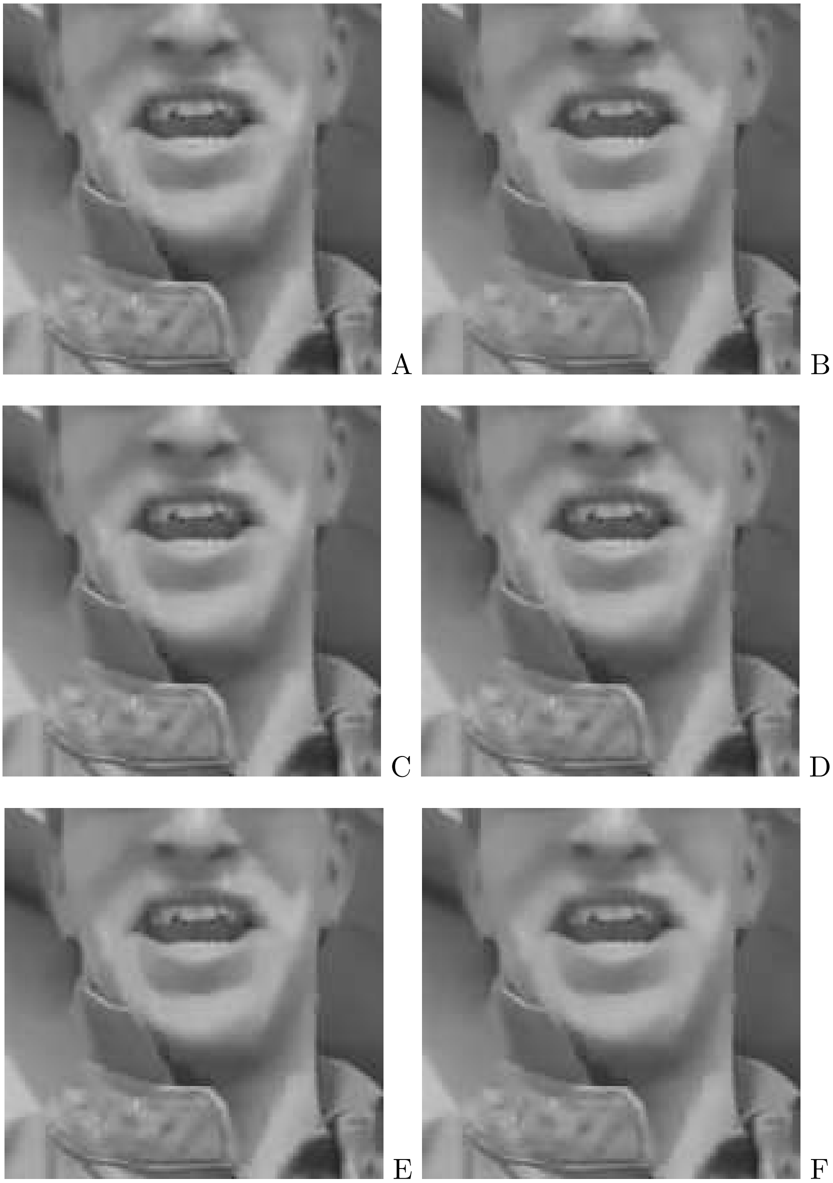
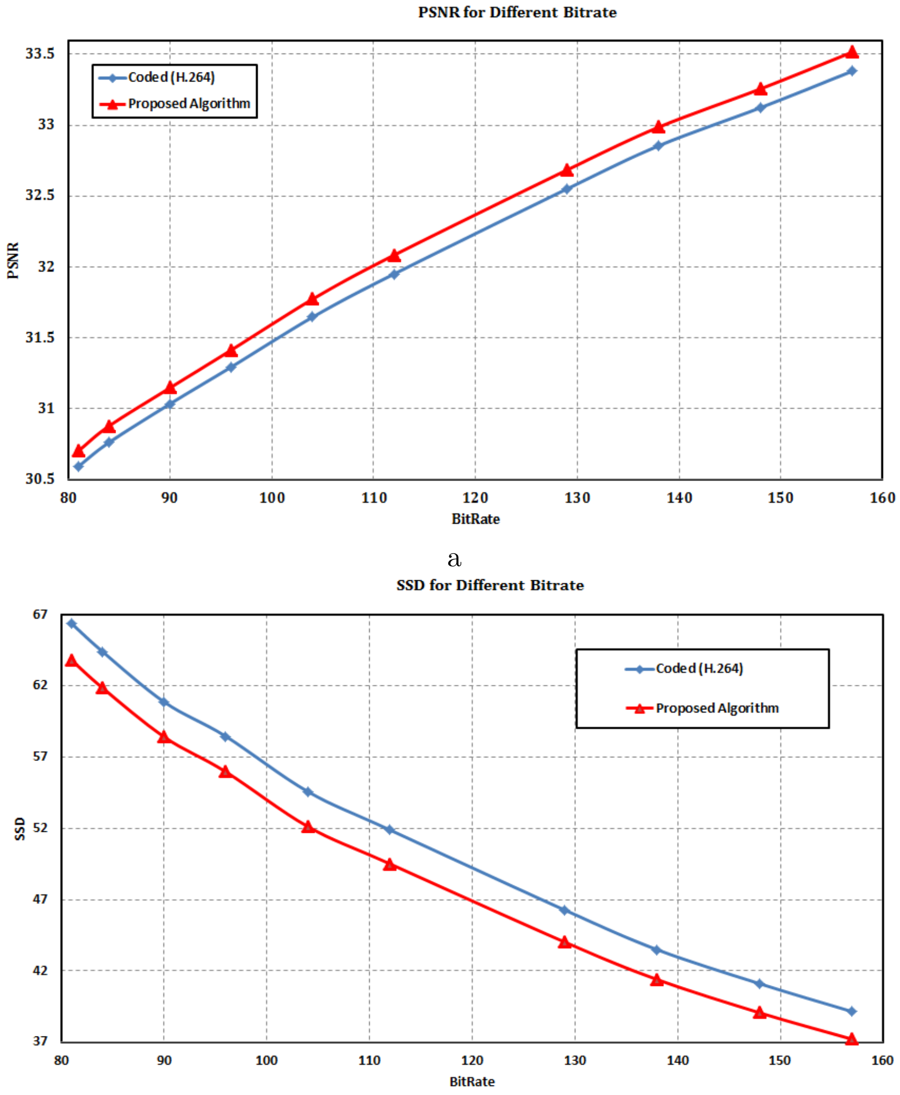


Figure 2.22: Comparison of filtered results for H.264/AVC sequences (56_{th} frame of Foreman zoomed): (A) the compressed H.264 with QP = 35 (PSNR=32.585, MSSIM=0.881); (B) Tai [47] (PSNR=32.451, MSSIM=0.887); (C) Zhai [27] (PSNR=32.610, MSSIM=0.881); (D) MCSTF [45] (PSNR=32.520, MSSIM=0.882); (E) Zhang [57] (PSNR=32.771, MSSIM=0.882); (J) the proposed method (PSNR=32.818, MSSIM=0.883).

Table 2.9: The average Y-SSD results for all the test video sequences using the selected algorithms (The sequences are 1:*Akiyo*, 2:*Akiyo*, 3:*Coastguard*, 4:*Container*, 5:*Cycling*, 6:*Foreman*, 7:*Hall*, 8:*Mobile* and 9:*Mother*, respectively.

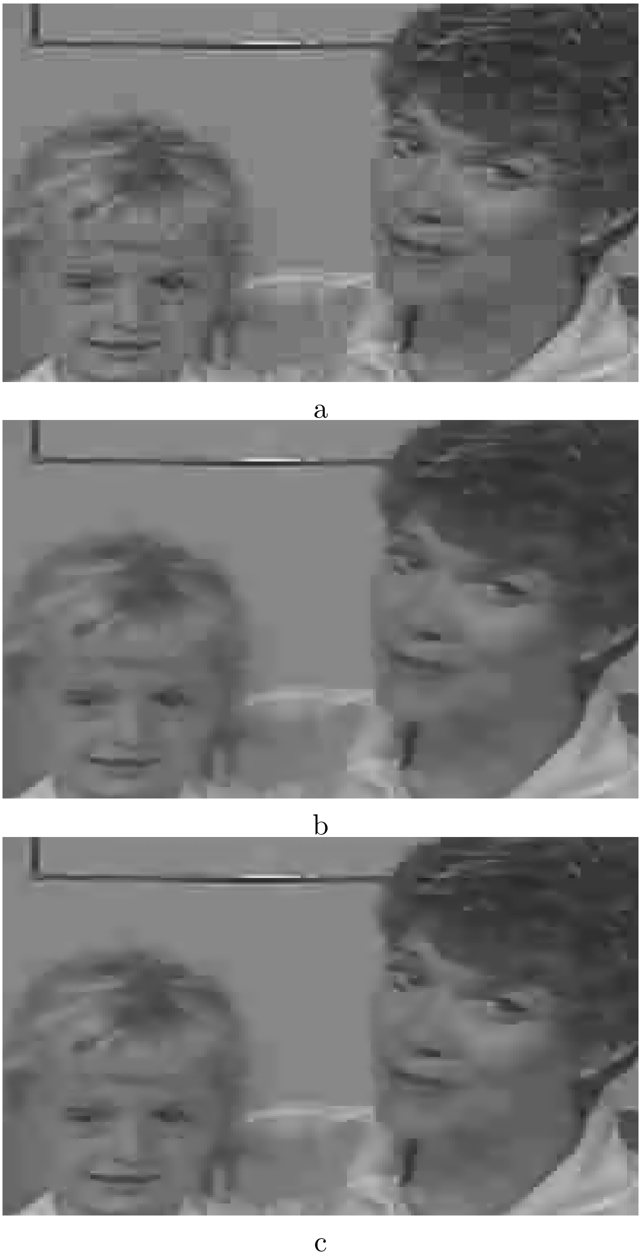
X	H.264/AVC		Vo et al.[45]		Bilateral [3]		Trilateral	
QP	35	45	35	45	35	45	35	45
1	4.68	10.84	4.37	10.08	4.46	9.95	4.58	10.29
2	111.27	416.58	124.61	416.31	110.34	412.43	110.01	409.31
3	89.05	192.78	90.73	189.33	89.01	188.31	88.39	190.68
4	6.93	17.26	6.80	16.19	6.71	16.17	6.79	16.36
5	66.49	214.62	72.42	211.31	65.15	211.75	65.11	212.35
6	38.02	89.59	38.47	85.84	35.60	84.77	34.24	84.02
7	15.82	33.74	15.66	31.97	15.47	32.09	15.33	32.05
8	98.39	254.34	101.93	246.93	94.13	248.44	93.68	248.37
9	8.46	15.64	8.23	14.86	8.34	15.03	8.22	15.40



a

b

Figure 2.23: Comparison of Y-PSNR and Y-SSD with different bit-rates of the *Foreman* sequence (The coded sequences and proposed sequences are the H.264 video with GOP = $(IPPB)_{12}$ and processed using algorithm explained in Section 2.5).



a

b

c

Figure 2.24: Comparison of filtered image using the proposed method and the last algorithm [3] on JPEG compressed image-4Q (a) Compressed image b) Filtered image using the bilateral-based method [3] c) Filtered image using the trilateral-based method.

2.6 Conclusion

Post-processing of coded images and video can reduce typical artifacts, both spatial and temporal. Spatial artifacts include blocking and ringing, whereas typical temporal artifacts are mosquito and flickering. Three methods to remove these artifacts have been presented in this Chapter.

The first method combines the strengths of anisotropic diffusion and fuzzy filtering. Anisotropic diffusion is effective in removing ringing noise, while fuzzy filtering can remove blocking. The spread parameter of the fuzzy filters are selected adaptively, depending on the activity of each block. Experimental results show that this proposal can successfully reach higher objective measures and better visual appearance in comparison to other algorithms.

The second method, designed for video sequences coded with the H.264/AVC standard, applies a directional deblocking filter with adaptive strengh, again depending on the characteristics of the blocks. Deringing is then achieved with an adaptive bilateral filter. Experimental results show that this method can beat the H.264/AVC in-loop filter and achieve higher PSNR and MSSIM than other similar algorithms.

The third method introduced a new filter for removing artifacts; an adaptive trilateral filter for removing spatial and temporal artifacts. The proposed algorithm is non-iterative and its complexity is low. In addition, it does not require motion estimation and compensation steps for post-processing. The experimental results showed that the proposed algorithm effectively reduced blocking, ringing and flickering artifacts. It outperformed other methods in terms of PSNR, MSSIM and SSD.

Chapter 3

LCD Backlight Dimming

This chapter is dedicated to the local backlight dimming and flickering artifact in LED-LCD display. Local backlight dimming can reduce the power consumption, and simultaneously, increase the contrast ratio. When local backlight dimming is applied over time, the flickering effect may appear. Section 3.1.4 is based on the work in [4] and the work in [5] is presented in Section 3.2.

A brief overview of local backlight dimming is presented and the State of the Art backlight dimming algorithms is reviewed and implemented in the first part of this chapter. Then, a new dimming algorithm is presented.

However, flickering appears in local backlight dimming over the time. The second part of this Chapter is about flickering artifact removal in local backlight displays. A new algorithm for flicker reduction is presented.

The reader can find more information regarding photometry, colorimetry principles and backlight in Liquid Crystal Displays (LCDs) display in [66].

3.1 LED-LCD Backlight Display

LCDs is one of the most preferable and popular display technology that is used in televisions, computers and in some projector systems. LCDs is nowadays the most widespread display type, that is used in many devices, from mobile phones to 3D TV. The global market of the flat panel display in 2008 was approaching US $ 100 billion [67]. Among this, LCDs occupied a 93% share [67]. Most LCD devices require backlight units. There are three important issues for developing backlight units: first, to reduce the power consumption of the backlight, since a backlight unit uses 70 to 80% of the total power used in a LCD module; second is to decrease the cost; and finally, to establish and maintain the adequate supplies. This Section addresses the power consumption in LCD TV, particularly Light Emitting Diodes (LEDs) as the light source of backlight display [66].

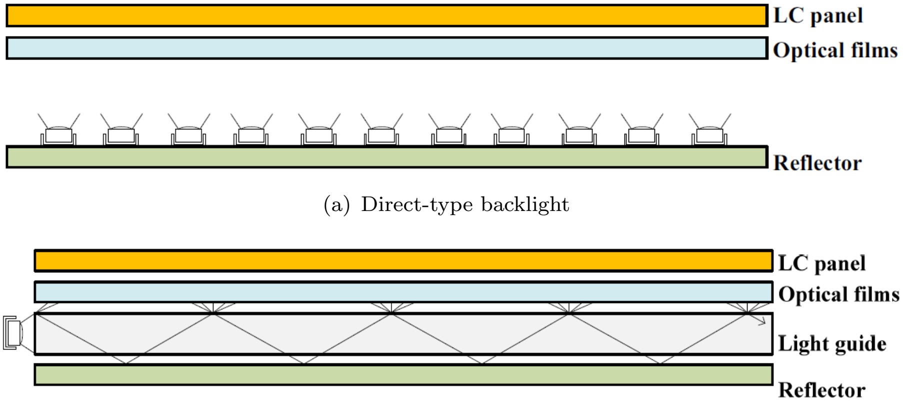
3.1.1 Background of Backlight Display

In this project, the focus was on the Full High Definition (HD) LCD television using LED- backlight, displaying images and video.

Since an Liquid Crystal (LC) is basically a voltage controlled light filter, an LCD requires a light source in addition to LC. 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 [67].

Based on the position of LEDs (light source) of an LED-LCD backlight display, they are called direct and edge type. In the direct type backlight, LEDs are placed in the back side of the optical films and LC panel (Fig. 3.1a) and in the edge type backlight, LEDs are placed along the edge of the display (Fig. 3.1b). The light is distributed from edges to the whole panel through light guide. Since the direct-type backlight has many numbers of local blocks, the efficiency of local dimming is higher than in the edge type backlight. However, the thickness of the edge type is less. This is because a certain level of distance between light source and optical films is required to achieve a uniform distribution of luminance. Figure 3.1 shows the structure of direct and edge type LED backlight.

In the past, Cold Cathode Fluorescent Lamps (CCFLs) were used in backlights. Today, they are replaced by Light Emitting Diodes (LEDs), thanks to the rapidly improving energy efficiency and decreasing price [22]. This also enables backlight dimming, because LEDs have shorter response time. There can be hundreds or even thousands of LEDs in one backlight unit. LED-based backlights can be divided into independently



(a) Direct-type backlight

(b) Edge-type backlight

Figure 3.1: Backlight display architectures of LED TVs

adjustable segments, called *local* backlights and *global* backlight, where a single segment covers the whole display; in this work, we use the word LED to refer to the 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.

Backlight dimming algorithms take the frame to be displayed as an input signal, and process it to return a signal that drives the backlight accordingly. Their goal is to reduce power consumption and increase the contrast. The contrast is the ratio between the luminance of the brightest emission of the screen and its darkest emission. Two kinds of contrasts can be defined in backlight display [66];

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

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

In some vendor specifications, dynamic contrast can be easily infinite, since it is sufficient to turn off the backlight to achieve such result. Static contrast is a more meaningful parameter and has a higher impact on image quality.

Backlight dimming algorithms can be classified similarly according to the backlight structures; depending on what the coarseness of the dimmed region is:

Global dimming (0D Dimming): In each frame, the backlight for all the pixels is dimmed by the same factor. In other words, the whole frame is considered as a single dimming region. This is a simple dimming method. However, it can only improve the dynamic contrast.

Local dimming: different regions of a frame are dimmed differently according to the content of the region itself. Local dimming can be further classified depending on how many dimming dimensions exist.

1-D dimming: The local dimming is performed only in one direction (bottom/up or left/right). 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 located at both sides so that the 1D backlight segments are split in two parts; this would be called 1.5D dimming.

2-D dimming: The local dimming is performed in two directions on a plane (the screen). Dimming regions can be identified by a 2-D spatial coordinates (X-Y). The single backlight segments are turned on, turned off or dimmed according to the content of the region they are enlightening.

RGB 2-D(3D) dimming: or color dimming, takes the advantage of colored backlight to adapt it according to the image.

Figure 3.2 shows some examples of the 0D, 2D and 1.5D backlight dimming.

When the backlight is divided into local segments, each one can be dimmed to match the brightness of its respective image area. This saves power, since dark areas require less light. Backlight dimming can also reduce *leakage*, a typical problem in LCDs: the LCs cannot completely block the light going through them. This makes the black pixels look grayish and consequently reduces the contrast ratio. Backlight dimming lowers the amount of light leaking through the dark pixels. The possibility to vary the backlight intensity also allows to increase the number of distinct luminance levels that can be emitted and as result, enables High Dynamic Range (HDR) rendering [66].

Several local backlight dimming algorithms were proposed in the literature [10], [16], [69]–[75]. In all of them, a trade-off was made between image quality, power saving and complexity. Several State of the Art algorithms are described in Section 3.1.2. In addition, the backlight dim-

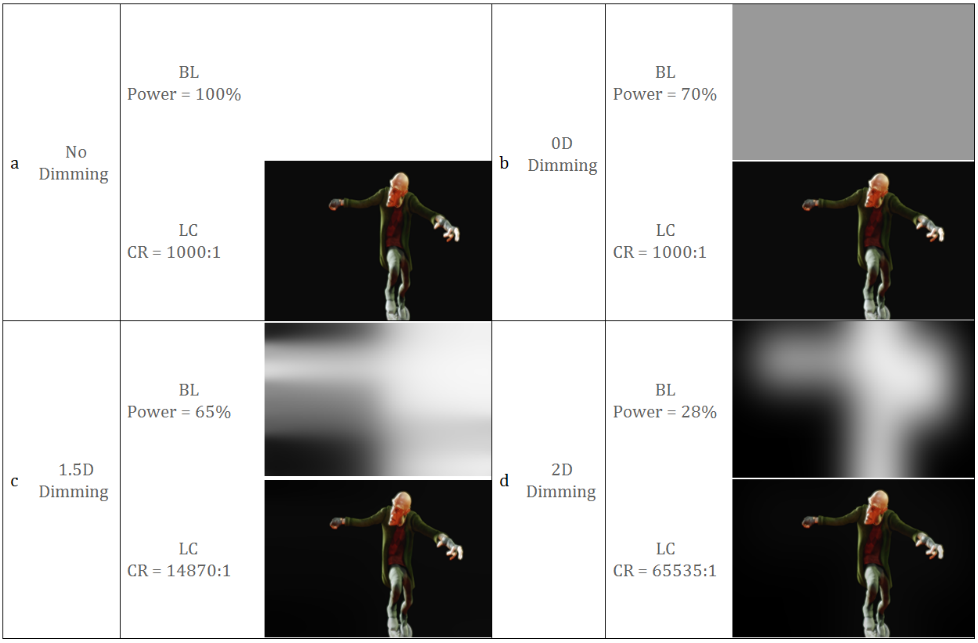


Figure 3.2: Effect of Dimming Methods on a image (CR: Contrast Ratio, D: Dimension, BL: Backlight, LC: Liquid Crystal). The contrast ratio is based on the Michelson contrast definition [68].

ming model, used in this Ph.D work will be presented in 3.1.3. Finally, the proposed method developed will be presented in Section 3.1.4.

3.1.2 State of the Art Dimming Algorithms

This section presents a selection of the State of the Art backlight dimming algorithms. These algorithms have been used for comparison against the algorithms developed in this Ph.D. work. For simplicity, we refer to these algorithms using the name of the first author of each publication.

The backlight dimming algorithms can be divided into two main categories:

Feature-Based Algorithms: The most common algorithms for calculation of a backlight luminance are based on image processing methods. In these algorithms, the backlight luminance is determined by using the characteristic data of a target image like: average, maximum or histogram. Several algorithms have been proposed [70], [72], [74]–[83].

Optimization-Based Algorithms: For a better consideration of the physical characteristics of a display, some algorithms have been presented for local backlight dimming. These algorithms use the Point Spread Function (PSF) information to model the light diffusion more accurately than the feature-based algorithms which leads to more complex algorithms [10], [69], [71], [84].

3.1.2.1 Feature-Based Algorithms

The simplest backlight dimming algorithms use simple global or local image characteristics, for example the maximum or average pixel value, to determine the backlight luminance. 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 [76]. The Square-root (Sqrt) algorithm uses the square root of the normalized average pixel value, i.e. valued between 0 and 1 [77]. 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. Therefore, it causes the flickering, while the Avg tends to produce excessively dim backlights and it may decrease the brightness and lose the information in bright segments. The Sqrt is a trade-off between the two methods. However, it is also very simple.

Other algorithms use more advanced image statistics to obtain better backlight luminance estimation. In the following section, a selection of them is presented.

Cho algorithm:

The algorithm introduced by Cho et al. [78] uses the relationship between the average and the maximum luminance of an input image to calculate the backlight luminance by adopting an additional correction term. The backlight luminance is calculated by:

$$r_k = g_{avg,k} + corr, \tag{3.1}$$

where r_k is the backlight intensity of LED k and $g_{avg,k}$ is the average luma value of the pixels of the target image (y) in the segment k. The correction term $corr$ is calculated as:

$$corr = \frac{1}{2} \times \left(d + \frac{d^2}{2^n} \right), \tag{3.2}$$

where $d = (g_{max,k} - g_{avg,k})$, $g_{max,k}$ is the maximum luma value of the pixels in the segment k and n is the bit depth of y. A flowchart illustrating the functioning of the algorithm is shown in Figure 3.3.

In addition, Cho et al. introduced another local dimming algorithm which consists of several steps [79]. First, the histogram of each color in a local segment is generated. In addition, the variance of luminance as well as the contrast in the dimming images are calculated. Second, the backlight luminance of every local segment is determined based on the distortion of the dimming image. Then, it is modified by comparing the dimming level among the nearby local segments to reduce the influence of light diffusion from other segments. At the end, the backlight luminance of each segment is calculated according to the dimming rates of the local segments. Figure. 3.4 shows the flowchart diagram for finding the backlight luminance using Cho's algorithm [79].

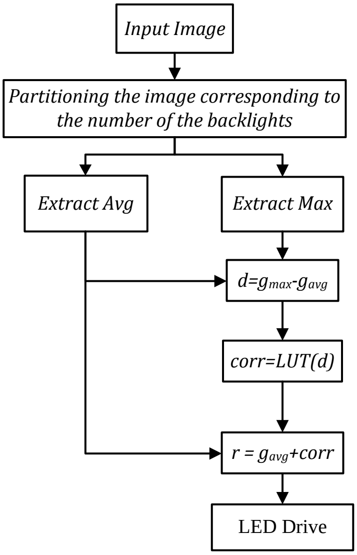


Figure 3.3: Flowchart of the Cho algorithm [78].

Zhang algorithm:

The algorithms introduced by Zhang et al. [80] use the same strategy of adding a correction term to the average luminance required by each segment as Cho's algorithm [78]. The correction term in Zhang algorithm [80] is calculated as follows:

$$corr = \begin{cases} 0 & \text{if } \sigma_k = 0 \\ \left(1 - \frac{\sigma_{avg}^2}{\sigma_k^2}\right) \times d & \text{if } \sigma_k \neq 0 \end{cases}, \tag{3.3}$$

where σ_k^2 is the variance of the luminance in segment k and σ_{avg}^2 is the variance when the maximum luminance of the backlight is equal to the average luminance of segment k. The main difference of Cho's and Zhang's algorithms is that, the second tries to compensate the loss of details that an excessively low luminance would cause, while the first corrects the backlight by considering the local difference between maximum and average luminance. The values of *corr* can be stored in a Look-Up Table (LUT) to make the algorithms faster.

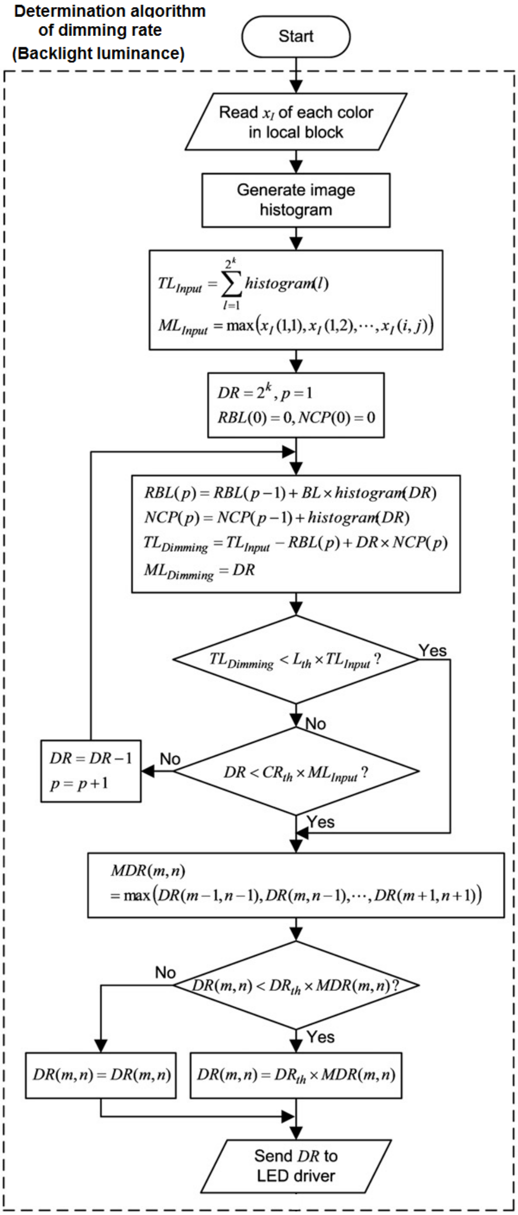

Determination algorithm of dimming rate (Backlight luminance)
Start
Read x_l of each color in local block
Generate image histogram
$TL_{Input} = \sum_{l=1}^{2^k} histogram(l)$
$ML_{Input} = \max(x_l(1,1), x_l(1,2), \cdots, x_l(i,j))$
$DR = 2^k, p = 1$
$RBL(0) = 0, NCP(0) = 0$
$RBL(p) = RBL(p-1) + BL \times histogram(DR)$
$NCP(p) = NCP(p-1) + histogram(DR)$
$TL_{Dimming} = TL_{Input} - RBL(p) + DR \times NCP(p)$
$ML_{Dimming} = DR$
$TL_{Dimming} < L_{th} \times TL_{Input}$?
Yes
No
$DR < CR_{th} \times ML_{Input}$?
No
$DR = DR - 1$
$p = p + 1$
Yes
$MDR(m,n)$
$= \max(DR(m-1,n-1), DR(m,n-1), \cdots, DR(m+1,n+1))$
$DR(m,n) < DR_{th} \times MDR(m,n)$?
No
Yes
$DR(m,n) = DR(m,n)$
$DR(m,n) = DR_{th} \times MDR(m,n)$
Send DR to LED driver

Figure 3.4: The second algorithm proposed by Cho et al. [79] to calculate the backlight luminance (The dimming rate is equal to the backlight luminance).

Nam algorithm:

The algorithm introduced by Nam [85], uses a combination of the global and local characteristics in an image. Depending on the local and global maximum and average luma values, the LEDs are set as follows:

$$r_k = \begin{cases} \left(\frac{g_{m,k}}{2^n-1}\right)^\gamma & \text{if } g_{m,k} \geq g_m \\ \left(\frac{g_{max,k}}{2^n-1}\right)^\gamma & \text{if } (g_{m,k} < g_m) \wedge (g_{max,k} \leq g_{avg}) \\ \left(\frac{g_{avg}}{b}\right)^\gamma & \text{if } (g_{m,k} < g_m) \wedge (g_{max,k} > g_{avg}) \end{cases}, \tag{3.4}$$

where $g_{m,k} = \frac{g_{max,k}+g_{avg,k}}{2}$, γ 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

$$b = (2^n - 1) \times \left(1 - \frac{1 - \frac{g_{avg}}{g_m}}{g_{max} - g_{avg}}(g_{max,k} - g_{avg})\right). \tag{3.5}$$

The author also described an algorithm-specific way to compensate the LCs for the light loss due to dimming. More details about the algorithm can be found in [85].

Kim algorithm:

Kim et al. [72] proposed an iterative local backlight dimming algorithm considering the characteristics of neighboring image segments to set the intensity of each LED. The authors defined two measures for severity of leakage and clipping artifacts in each block. The algorithm iteratively calculates the backlight luminance 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. The iterative algorithm is described in algorithm 2. More details about the clipping and leakage error measures, the weights and the iterations can be found in [72].

Chen algorithm:

Chen et el. [70] proposed a new method for backlight dimming, namely pixel-compensated backlight dimming (PCBD). Figure 3.5

Algorithm 2 The Kim Algorithm [72], α_k is the weight parameter, L_m and C_m are leakage and clipping measures, respectively.

```
Step 1: initialize $r_{max} = 255, r_{min} = 0, r_k = g_{avg,k}$;
Step 2: calculate $\alpha_k$
Step 3: calculate $L_m$ and $C_m$ using $r_k$.
Step 4:
if $L_m > \alpha_k \times C_m$ then
  $r_{max} = r_k, r_{min} = r_{min}$
else
  $r_{min} = r_k, r_{max} = r_{max}$
end if
Step 5: update $r_k = \lceil (r_{max} + r_{min})/2 \rceil$.
Step 6: repeat step 3 - step 5 until $r_{max} - r_{min} = 1$
```

shows the structure of the PCBD algorithm which comprises three steps: pre-processing, local dimming, and global dimming. In the pre-process step, the input image is smoothed by a low-pass filter to reduce the disturbance of noise when calculating the intensity of each backlight block. To avoid saturation and overflow, the pixel gray level is extracted as the maximal value among the RGB of each pixel. The local dimming step includes backlight local dimming and local pixel compensation. In the backlight local dimming, the luminance intensity of each LED block is calculated by analyzing the gray level histogram of the correspondent image block, and then the pixel values are locally compensated according to the backlight optical profile. In this step a temporal filter was proposed to eliminate screen flicker which will be discussed in flickering reduction Section. The global-dimming step includes backlight global dimming and global pixel compensation. The global dimming part tries to minimize the backlight-dimming potential in the bright-image region at the precondition of no brightness lost. More details about the algorithm can be found in [70].

Wang algorithm:

Wang et al. [81] proposed a modulated algorithm to determine the RGB-backlight luminance. The algorithm uses a delta-color adjustment (DCA) to optimize the backlight image. The flowchart

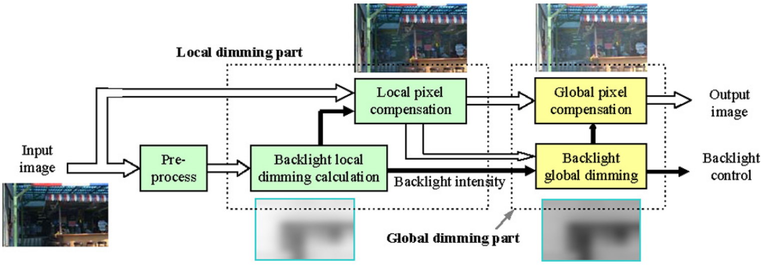


Figure 3.5: Structure of the locally pixel compensated backlight dimming (PCBD) algorithm proposed by Chen et al. [70].

of DCA algorithm is shown in Figure 3.6. First, the backlight luminance of each image is determined by any backlight-control method in phase 1 such as the IMF [74], Max or Square root method. Then, the backlight luminance (BL) is divided into three equal gray level (GL) of three-dimension BL (BL(R), BL(G), and BL(B)). Next, the R, G, and B sub-pixel values in each backlight segment are averaged for three-dimension (R, G, and B) BL individually to find the reference values. Finally, the references are used to find three delta BL values (ΔBL_R, ΔBL_G, ΔBL_B) to determine the optimized color-backlight luminance.

The key point of the DCA method was to obtain the optimized ΔBL values [81]. First, the BL value ($0 \sim 255$) is divided into equally M parts to map the new values ($1 \sim M$) as shown in Figure. 3.7. Second, in order to obtain a new mapping value ($M_{intensity}$), the conventional luminance control method issued to detemine a new backlight luminace using the initial backlight luminance, is calculated in Phase 1. Third, the average gray level of R, G, and B pixels in each backlight segment is calculated individually, and used in a look-up table to map three reference values ($R_{r,g,b}$). Fourth, a threshold value N is adjusted, as shown in Eq. 3.6. If the three transient BL values ($T_{r,g,b}$) are larger than N, the RGB backlight signals will be dimmed, otherwise the same intensity control values from Phase 1 will be kept. Finally, $256/M$ is multiplied by the three delta values ($D_{r,g,b}$) individually to calculate three delta BL values ($\Delta BL_{R,G,B}$), as shown in Eq. 3.7.

$$T_{r,g,b} = M_{intensity} - R_{r,g,b}$$
$$D_{r,g,b} = \begin{cases} M_{intensity} - R_{r,g,b} - N & \text{if } T_{r,g,b} > N_{Threshold} \\ 0 & \text{otherwise} \end{cases} \tag{3.6}$$

$$\Delta BL_{R,G,B} = D_{r,g,b} \times 256/M, \tag{3.7}$$

Different M (division adjustment) and N (threshold adjustment) values will influence image quality and power consumption in the DCA method.

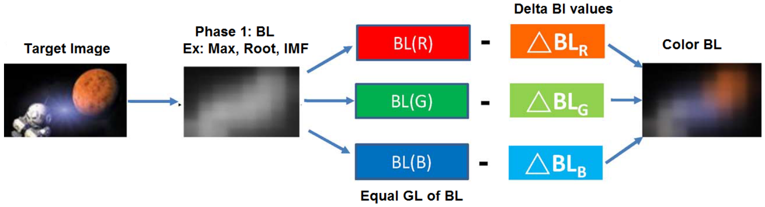
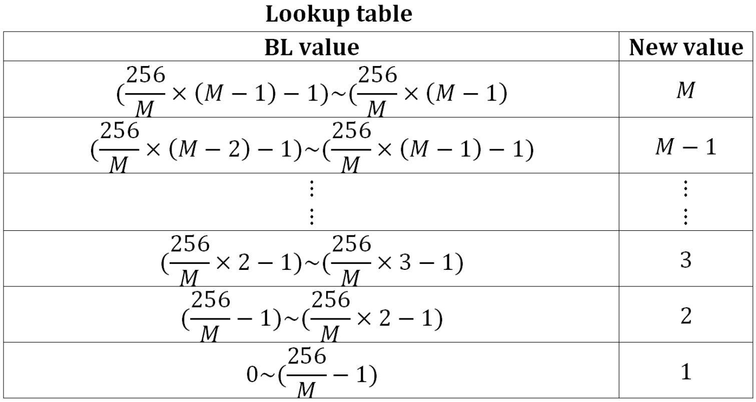


Figure 3.6: Flowchart of the Delta-Color Adjustment (DCA) algorithm proposed by Wang et al. [81].



Lookup table

BL value	New value
$(\frac{256}{M}\times(M-1)-1)\sim(\frac{256}{M}\times(M-1)$	M
$(\frac{256}{M}\times(M-2)-1)\sim(\frac{256}{M}\times(M-1)-1)$	$M-1$
⋮ ⋮	⋮ ⋮
$(\frac{256}{M}\times 2-1)\sim(\frac{256}{M}\times 3-1)$	3
$(\frac{256}{M}-1)\sim(\frac{256}{M}\times 2-1)$	2
$0\sim(\frac{256}{M}-1)$	1

Figure 3.7: Lookup table of the DCA method [81].

Lin algorithm:

Some backlight algorithms use more features of the image, such as local or global histogram of the pixel values [74], [79], [82]. In [74], Lin et al. introduced a local dimming algorithm using the inverse mapping function. In the first step, a zone-weighting value of each backlight segment is decided, and then, the global histogram of the target image is computed to get a Probability Density Function (PDF). Then, the mapping function is calculated using the cumulative distribution function (CDF). Finally, inverse of the mapping function (CDF) of the target image with the oblique line ($y = x$) is calculated to generate a new mapping function for backlight modulation, named "Inverse of a Mapping Function (IMF)" [74]. The weighting $n(0 \leq n \leq 1)$ was taken for the average and maximum values in each backlight segment to optimize the image quality and power consumption. The zone value is calculated as:

$$\text{Zone-weighting Value} = n \times \text{Max} + (1 - n) \times \text{Avg}, \qquad (3.8)$$

where $n = 0.9$ is the optimization value for the IMF algorithm. A very significant feature of the IMF method is that it optimizes the backlight luminance signal according to each input image, i.e., backlight luminance with a dynamic gamma controlled frame by frame to produce high quality images in high and low contrast ratio images.

Kang algorithm:

Kang et al. [82] presented a multi-histogram-based algorithm for global dynamic backlight dimming. In Kang's algorithm, the input image is divided into blocks, each containing a backlight segment, to determine the multi-histogram. These multiple histograms are used to analyze the pixel distribution for RGB components of the input image. The backlight luminance is calculated using the total squared errors of the multiple histogram and maximum total square error of each block by trial and error iteratively [82]. The total squared errors are calculated using the inverse of the target PSNR. Two methods, based on the maximum value selection of the RGB and histogram binning are introduced to reduce the complexity of the algorithm [82].

As shown in figure 3.8(a), if the maximum total square error (TSE_M) is smaller than the target total square error (TSE_T), the maximum clipping point (I_{CCP}) is decreased by one and TSE_M should be computed again. If TSE_M is larger than TSE_T, the previous I_{CCP} should be selected as the backlight luminance.

S.I. Cho algorithm:

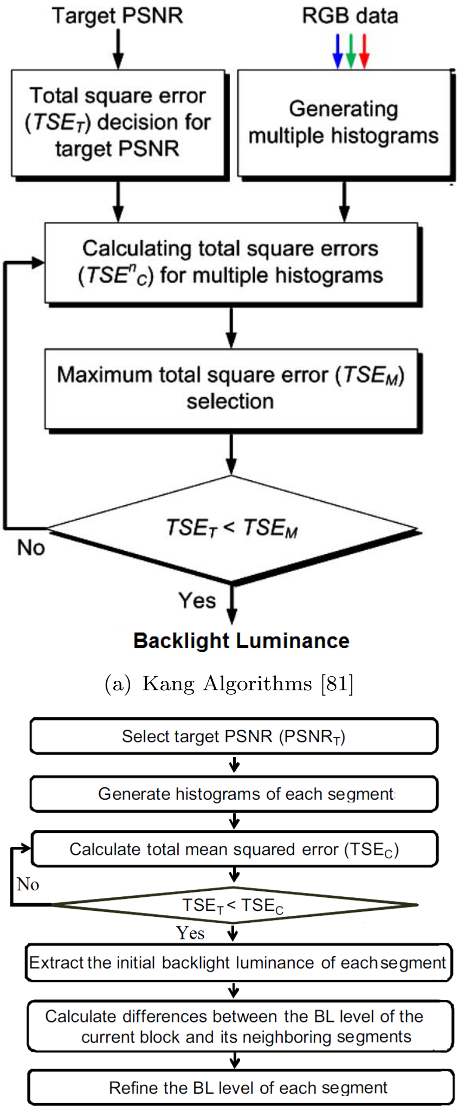
S. I. Cho et al. [83] presented another version of local dimming using histogram analysis. In that method, the backlight luminance level of each segment is determined in two steps. In the first step, the algorithm dynamically selects the initial backlight luminance level of each segment, considering the peak signal-to-noise ratio and histogram analysis, as in Kang's algorithm [81]. In the second step, the backlight luminance is refined by considering the light coming from neighboring segments, as well as the relations between the current segment and its neighboring segments. The flowchart in figure 3.8(b) shows the proposed algorithm by S. I. Cho et al. [83].

3.1.2.2 Optimization-Based Algorithms

Some algorithms use the information of the light diffusion in the backlight more accurately than others [69], [71], [84]. Therefore, these algorithms are more complex. The information of light diffusion is represented by the PSF, also called light spreading function [74]. It shows how the light spreads on the diffuser plate from a light source. Each backlight segment has a characteristic PSF. In the following section a selection of them is presented.

Albrecht algorithm:

Albrecht et al. [69] is considered the backlight luminance calculation of a local dimming as an optimization problem. The luminance produced by many LEDs for each pixel considered is calculated, which should cover the gray value of each pixel, while the sum of LED currents should be minimized. For this purpose a specific algorithm called as Sorted Sector Covering (SSC) was developed [69] (See figure 3.9).


Target PSNR
RGB data
Total square error (TSE_T) decision for target PSNR
Generating multiple histograms
Calculating total square errors ($TSE^n{}_C$) for multiple histograms
Maximum total square error (TSE_M) selection
$TSE_T < TSE_M$
No
Yes
Backlight Luminance
Select target PSNR ($PSNR_T$)
Generate histograms of each segment
Calculate total mean squared error (TSE_C)
No
$TSE_T < TSE_C$
Yes
Extract the initial backlight luminance of each segment
Calculate differences between the BL level of the current block and its neighboring segments
Refine the BL level of each segment

(a) Kang Algorithms [81]

(b) S.I.Cho algorithm [83]

Figure 3.8: Flowchart of Kang and S.I.Cho algorithms [81], [83].

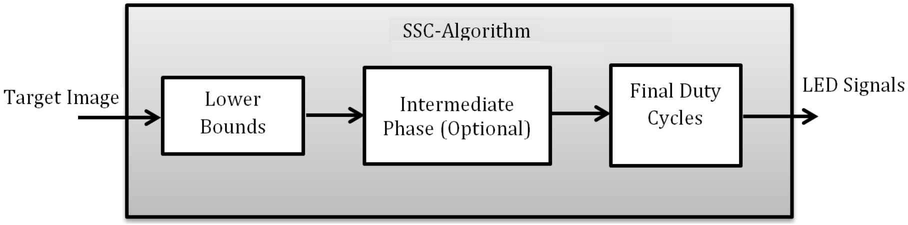


Figure 3.9: The algorithm proposed by Albrecht [69].

The algorithm 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: at each iteration, the pixel requiring the most additional luminance to reach to its target value is found. Then, the intensity of the most influential LED, not already at full power range, 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 the clipped pixels and the luminance of the corresponding LED is increased if any is found.

Hong algorithm:

The algorithm presented by Hong et al. [71] determines, for each backlight segment, a uniform luminance level which would keep a clipping measure below a certain threshold. Then, it finds the combination of LED intensities that keeps the power consumption to a minimum, while keeping the clipping below threshold in all segments. To do this, the algorithm uses a block-based coarse PSF, which assumes a uniform backlight inside one segment. This reduces accuracy as well as complexity.

Somehow similarly to [82], [83] and [71] use the 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 algorithm:

In [84], backlight dimming is modeled as an optimization problem. The goal is to minimize a cost function defined by the norm of the difference between the target physical luminance $\boldsymbol{y}$ and the actual physical luminance $\boldsymbol{x}$ emitted by the display pixels to obtain the maximum quality. The reduction of power consumption is a by-product of the quality optimization, due to the fact that leakage is included in the optimization, which pushes to reduce the backlight intensity. The output $\boldsymbol{x}$ is determined by the backlight $\boldsymbol{b}$ and by the pixel transmittances $\boldsymbol{t}$ [10]. The backlight depends on the influence matrix $\boldsymbol{H}$ and on the LED values $\boldsymbol{r}$. The values of $\boldsymbol{t}$ are within the leakage factor ε and 1, while those of $\boldsymbol{r}$ are limited between 0 and 1.

The formulation of the problem is the following:

$$\begin{array}{rl} \text{minimize} & \|(\boldsymbol{y}-\boldsymbol{x})\| \\ \text{subject to} & \boldsymbol{x}=\boldsymbol{b}\circ\boldsymbol{t} \\ & \boldsymbol{b}=\boldsymbol{H}\boldsymbol{r} \\ & \varepsilon \le t_i \le 1, \quad i=1,\dots,N \\ & 0 \le r_k \le 1, \quad k=1,\dots,M \end{array} , \tag{3.9}$$

where the $\circ$ operator defines element-wise multiplication. The norm to be minimized can be ℓ_1 or ℓ_2. The optimal r_k can be found using software solvers taking the problem and data as input [86], [87]. After optimization, the values of $\boldsymbol{r}$ result in the optimal backlight. More details about the algorithm and extended verions can be found in the papers [10], [15], [16].

3.1.3 Modeling

This section illustrates a mathematical model of LCDs which is used for developing of the local backlight dimming algorithms described in Section 3.1.2. In order to implement any backlight dimming algorithm, it is necessary to model the essential characteristics of the display with local backlight dimming capabilities. The most essential concepts are *transmittance* and *backlight intensity*; the transmittance defines the proportion of light that an LC pixel allows to go through; the backlight intensity defines the local brightness level of the backlight. These are the input signals to the LCD and determine the resulting image. When a color display is concerned, each pixel actually contain three subpixels: Red, Green and Blue, or RGB. The observed backlight intensity at a certain pixel position depends on the physical structure of the diffuser plate. The distribution of light coming from a single backlight element is modeled using a Point Spread Function (PSF), and the contributions from different light sources should be summed up to model the total observed backlight at each pixel. Because of light leakage, there may also be a mismatch between the intended transmittance and observed transmittance. All these aspects need to be considered in the backlight model [10].

Figure 3.10 illustrates a generic LCD with local backlight dimming. As shown in Figure 3.10 to display an image, the backlight dimming algorithm takes the input image and display parameters like light diffusion or leakage and calculates the LED values and pixel transmittance. The input image can be in a sRGB color format or normalized RGB values, using gamma correction with an exponent of 2.2 [10].

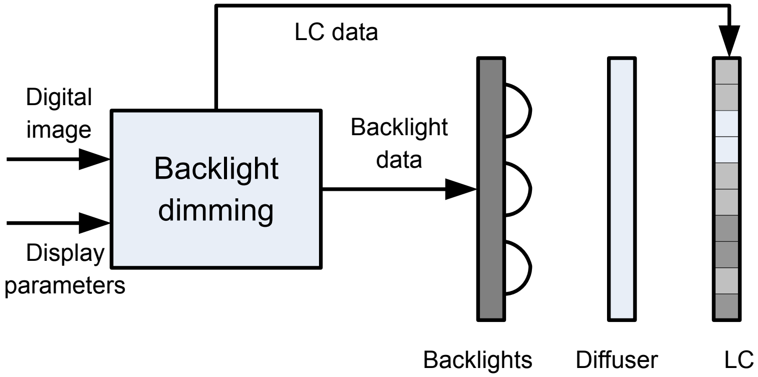


Figure 3.10: Structure of a LCD with local backlight [66].

3.1.3.1 Transmittance and leakage

In a locally backlight LCD screen, the observed pixel luminance l can ideally be expressed as the product of the backlight intensity b and the LC transmittance t [88]:

$$l = bt. \tag{3.10}$$

The values of l, b and t are normalized between $[0, 1]$. If $b = 0$ there is no light behind the pixel, while if $b = 1$ the light intensity is maximum. 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 [88]. Leakage can be modeled linearly by using a parameter ε, 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

$$l = bt + \varepsilon b(1 - t), \tag{3.11}$$

or alternatively:

$$l = bt_o, \tag{3.12}$$

where t_o is the observed transmittance, as opposed to ideal transmittance t; t_o can be expressed as

$$t_o = (1 - \varepsilon)t + \varepsilon. \tag{3.13}$$

The model, Equations 3.11-3.13, allows each pixel to have a different ε value, as leakage increases with the viewing angle θ [66]. 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 ε across the screen.

3.1.3.2 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

$$b = \textstyle\sum_{k=1}^{N} r_k h_k, \tag{3.14}$$

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, Equation. 3.14 for all pixels is

$$\boldsymbol{b} = \boldsymbol{H}\boldsymbol{r}, \tag{3.15}$$

where the column vector $\boldsymbol{b}$ has a b value per pixel, the *influence matrix* $\boldsymbol{H}$ (N columns and a row per pixel) represents the PSFs and $\boldsymbol{r}$ 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 [71]; 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. 3.11). 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 [66]. 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.

3.1.3.3 Brightness compensation

The reduction of luminance caused by dimming the backlight can be compensated by increasing the pixel transmittance. This is called *brightness compensation* [89]. In an ideal case, the compensated transmittance t_C can be solved from the equation. 3.10 by replacing l with the target pixel luminance l_y:

$$t_C = \frac{l_y}{b}. \tag{3.16}$$

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 ε due to the leakage.

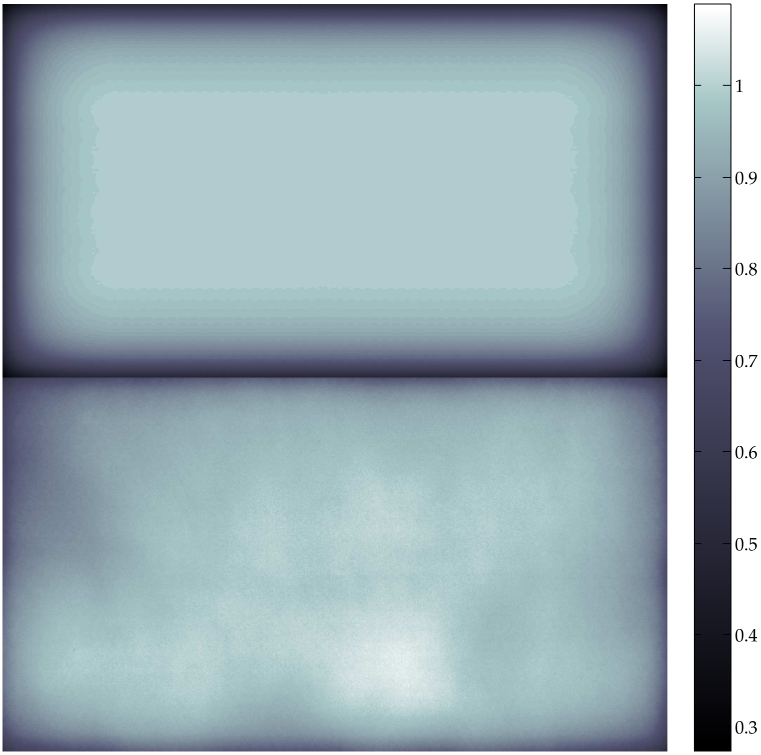


Figure 3.11: Example of modeled (top) and measured (bottom) backlight. Here, the modeled backlight is obtained by applying a single PSF on many segments, causing dim edges and non-accurate representation of the actual backlight, one example of which is shown in the bottom part [66].

The equation 3.16 should thus be rewritten as

$$t_C = \begin{cases} 1 & \text{if } l_y/b > 1 \text{ (Cond. I)} \\ 0 & \text{if } l_y/b < \varepsilon \text{ (Cond. II)} \\ (l_y/b - \varepsilon)/(1-\varepsilon) & \text{otherwise} \end{cases} . \tag{3.17}$$

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 the compensated pixels are brighter than intended. The observed physical transmittance can be computed from equation. 3.13, by replacing t with t_C.

It should be noted that equation. 3.17 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 [89]. Equations 3.10-3.17 can be applied for white backlight and a monochrome LC signal. When more color components are used, as in RGB LCD [81], 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. More information is described in [66].

3.1.3.4 Quality Assessment of Algorithms

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 $\boldsymbol{y}$ and the output $\boldsymbol{x}$ of the algorithm in the physical or in the perceptual domain [66]. Common measures include the Mean Absolute Error (MAE):

$$\mathrm{MAE} = \frac{\sum_{i=1}^{N} |y_i - x_i|}{N}, \tag{3.18}$$

the MSE:

$$\text{MSE} = \frac{\sum_{i=1}^{N}(y_i - x_i)^2}{N}, \tag{3.19}$$

and the PSNR, which is built on MSE:

$$\text{PSNR} = 10 \times \log_{10}\left(\frac{\text{MAX}^2}{\text{MSE}}\right), \tag{3.20}$$

where N is the number of pixels in the screen and MAX is the maximum pixel value. For non-normalized 8-bit sRGB values, MAX = 255. In this work normalized pixel values were used, therefore MAX = 1 unless differently stated.

To validate the backlight dimming model, the visual quality of the backlight dimming algorithms were tested using subjective quality assessment [4].

3.1.3.5 Equipment

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

The first screen is the model of a real 47 inch Full High Definition (HD) display manufactured by SIM2 [90]. The backlight has 2202 segments arranges in hexagonal grids. All segments use the same PSF, provided by the manufacturer. An example of backlight generated by this screen is shown in the bottom-left corner of Figure 3.2.

The second screen is edge-lit and has 16 segments, placed in 8 rows and 2 columns. Similar to 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 3.2

All screens have a default resolution of 1920 × 1080 pixels and it is possible to define a downscaling factor that shrinks and resizes the displays and the PSFs accordingly. Different factors can be specified for the horizontal and vertical dimensions.

3.1.4 Local Feature-Based Algorithm

This section introduces three new local feature-based algorithms. They use some characteristics of the target image, such as histogram, average and maximum pixel value of each segment, to reduce power consumption of the backlight and improve the image quality. In the following section, they will be presented.

In the first algorithms, a local histogram analysis is used to determine the backlight level. It consists of three steps, detailed below. The input RGB images are assumed to have a bit depth of 8 bits.

In the first step, the RGB values are normalized and multiplied by 255. Then, the average sub-pixel value (in the three RGB channels) avg_k is calculated for each backlight segment k. Each segment is categorized into a $class_k$ depending on avg_k:

$$class_k = \begin{cases} \text{low} & \text{if } 0 \leq avg_k \leq 31 \\ \text{medium} & \text{if } 32 \leq avg_k \leq 95 \\ \text{high} & \text{if } 96 \leq avg_k \leq 255 \end{cases} \tag{3.21}$$

In the second step, the local histogram of the pixels within each backlight segment is calculated. In order to reduce the light leakage, a quantile analysis of the histogram is used to initialize the backlight luminance. To reduce the calculation load for the histogram, the gray level of each pixel is defined as follows:

$$g = max(R, G, B) \tag{3.22}$$

Then, the Initial Backlight Level (IBL) for the segment k is defined. It is set to the lowest intensity that is higher than $\phi\%$ of the linearized g values in the segment (Figure 3.12). The relation between IBL$_k$ and ϕ is defined by the function $IBL_k(\phi)$.

The third and final step uses $class_k$ and IBL$_k$ to set the LED values:

$$r_k = \begin{cases} P(IBL_k(60), avg_k) & \text{if } class_k = \text{low} \\ P(IBL_k(65), sqrt_k) & \text{if } class_k = \text{medium} \\ P(IBL_k(70), sqrt_k + cv_k) & \text{if } class_k = \text{high} \end{cases}, \tag{3.23}$$

with

$$P(IBL, x) = \begin{cases} IBL + \alpha \times x & \text{if } 0 \leq max_k - avg_k \leq 31 \\ IBL + \beta \times x & \text{if } 32 \leq max_k - avg_k \leq 95 \\ IBL + \gamma \times x & \text{if } 96 \leq max_k - avg_k \leq 255 \end{cases}, \tag{3.24}$$

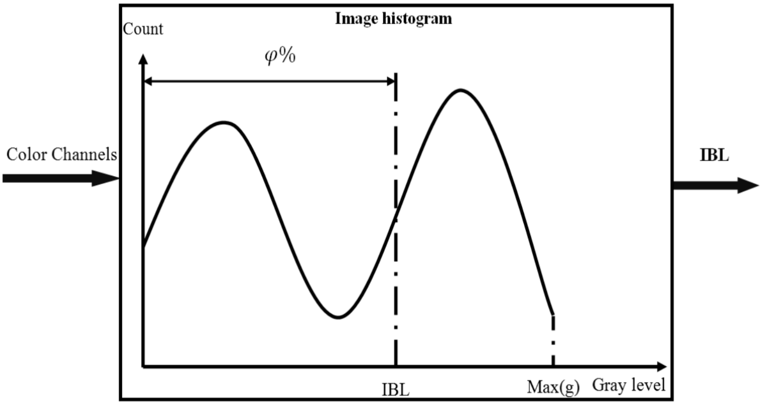


Figure 3.12: Calculation of the initial backlight value by histogram percentile [4].

where max_k is the maximum pixel value in segment k and α, β and γ are the parameters of the algorithm. $sqrt_k$ is the square root of the normalized avg_k multiplied by 255. The term cv_k is the ratio between the variance of the pixels in segment k and their average; it is a correction term that increases the backlight intensity in segments of class high. In fact, this algorithm combines the Avg and Sqrt algorithms [77] to apply them adaptively depending on the segment characteristics.

In the second algorithm, the IBL for the segment k is set to $\phi \times max_k$ of the linearized g values in the segment. The relationship between IBL_k and ϕ is defined by the function $IBL_k(\phi)$. The LED values are calculated as follows:

$$r_k = \begin{cases} P(max_k(\phi), avg_k) & \text{if } class_k = \text{low} \\ P(max_k(\phi+5), sqrt_k) & \text{if } class_k = \text{medium} \\ P(max_k(\phi+10), sqrt_k + cv_k) & \text{if } class_k = \text{high} \end{cases}, \qquad (3.25)$$

where $cv_k = \sigma_k^2 / avg_k$.

The third algorithm uses the two simple feature-based algorithms (Avg and Max) to calculate the LED values. First, the g values of the segment k are linearized and then, the IBL for that segment is calculated as $\phi \times max_k$ which is similar to the second algorithm. The LED values are determined as follows:

$$r_k = IBL_k(\phi) + \alpha \times avg_k, \qquad (3.26)$$

The third algorithm was used for objective and subjective evaluations and it will be refereed as 'proposed algorithm'.

3.1.4.1 Experiments

To evaluate the performance of the proposed algorithm against the other backlight algorithms, objective and subjective experiments were conducted. The objective evaluation compares the results of the proposed version with 11 algorithms from the literature (As described in 3.1.2) over 32 images while, for practical reasons, the subjective evaluation compares it with 4 of the other algorithms on 7 images. The two evaluation processes are described in Sections 3.1.4.2 and 4.2, respectively. All the algorithms were implemented in Matlab.

3.1.4.2 Objective Evaluations

The simulations were run on 32 different images (shown in Figures B.1 and B.3 in Appendix B). All images were rescaled to Full HD resolution with Bicubic interpolation (1980 × 1080). For the simulation, first the input RGB images were normalized with the standard Gamma function, $\gamma = 2.2$. The resulting backlight intensity values were calculated in physical domain as an 8 bit depth image. Then, backlight luminance values were normalized between 0 and 1 to calculate the relative power. Finally, the inverse Gamma function was applied to the normalized result to return to a perceptually uniform representation. The reported MSE results are based on the normalized values in perceptual domain [4].

The proposed algorithm was run on the edge-lit and SIM2 screen descried in Section 3.1.3.5 with $\varepsilon = 0.001$ and $\varepsilon = 0.0002$. The values of α and ϕ, were set to 0.8 and 0.6, respectively. After applying dimming algorithms, hard clipping in pixel compensation was performed, as described in 3.1.3.3.

The power consumption of the backlight was calculated using the average of LED values equation (3.14). MSE and PSNR were used to evaluate the resulting image quality in the perceptual domain.

Some backlight dimming algorithms were selected to compare our approach against other other algorithms. These algorithms are: the SQRT (square root of the average pixel value used for the backlight), the Max (maximum pixel value used for the backlight) presented in [77], Cho et al. [78], Wang et al. [81], Zhang et al. [80], Kim et al. [72], Lin et al. [13] [74], Chen et al. [70], Nam [85], Cho et al. [79], Kang et al. [82], S. I. Cho et al. [83] and Albrecht et al. [69]. For brevity, we have named these algorithms according to the first author name. For the Albrecht algorithm, the normalized RGB values were linearized with the standard Gamma function and in addition, two of the three steps were implemented, which produce a clipper-free result. There are two Cho algorithms, in this work named Cho [78] and Cho13 [79] for the first and the second algorithm, respectively. Since Cho13 was a color local dimming algorithm, and in our implementation the color backlight was not used, the target image was converted to a gray level image. Kang's algorithm is a global dimming algorithm, and it was generalized to local dimming. In the generalized form, the number of blocks in each backlight segment was set to four [82]. Three different versions of Kang algorithms were implemented, the first one is the global algorithm (global Kang), the second and the third are the low target PSNR (set to 30 db) denoted as Kang Low, and the high target PSNR (set to 50 dB) denoted as Kang High, respectively. All the algorithms were applied to both SIM2 and edge-lit displays, except S.I. Cho, that was only applied to the edge-lit display. The results of Chen and Nam algorithms were removed from these figure, because they were very low power algorithms.

Figures 3.13- 3.14 show average MSE results. On both displays, the proposed algorithm is among those with the lowest MSE and the power consumption was moderate. As shown in Figure 3.13, the proposed algorithm has the best performance in terms of quality, when ε is 0.001. In general, particularly considering its low complexity, the proposed algorithm can provide high quality results with at a reasonable power consumption.

To use the LED and LC values on real screens, they need to be quantized. For the backlight intensity, quantization has a negligible impact, but the LC quantizing has a significant impact on the numeric quality [10]. In Figure. 3.15, the results for LC values quantized to 8 bits are shown. The SQRT, Kang's algorithm (low), Kim's algorithm, Cho's algorithms reduce the power consumption of the backlights. However, the resulting image qualities are worse than with the other algorithms. With the Max, Zhang's, Kang's algorithm (high), Albrecht's, S.I. Cho's algorithm, the new Cho's algorithm (Cho13), Wang's and Lin's algo-

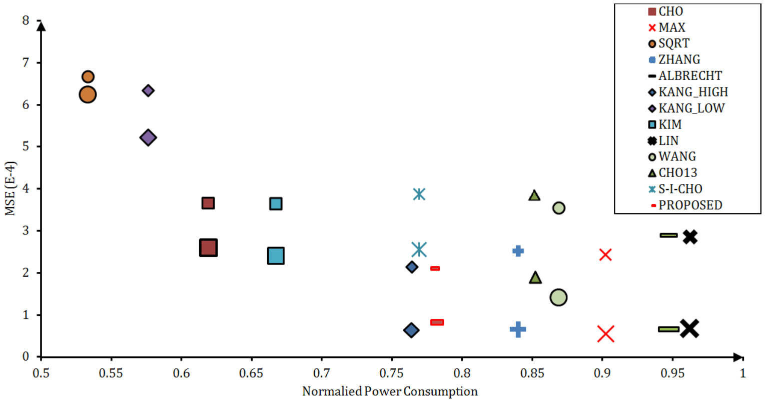


Figure 3.13: MSE/power of the proposed algorithm using local histograms compared with other algorithms on edge-lit displays. The smaller markers indicate $\varepsilon = 0.001$, the bigger ones $\varepsilon = 0.0002$ [4].

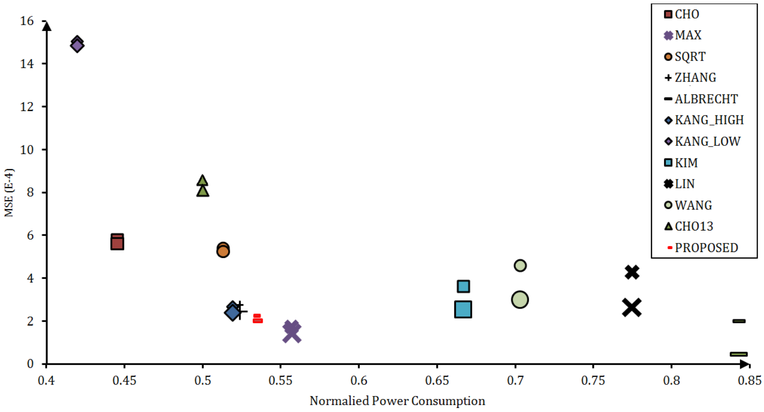


Figure 3.14: MSE/power of the proposed algorithm using local histograms compared with other algorithms on direct-lit SIM2 display. The smaller markers indicate $\varepsilon = 0.001$, the bigger ones $\varepsilon = 0.0002$ [4].

rithms, the image quality is good. However, the backlight power saving is small. With the proposed algorithm, image quality and backlight power savings for the various images are better balanced compared with the others. As shown in the figures, the backlight algorithms can be divided in two different categories: low power algorithms and high quality algorithms. The proposed method provides a good compromise between the image quality and power consumption.

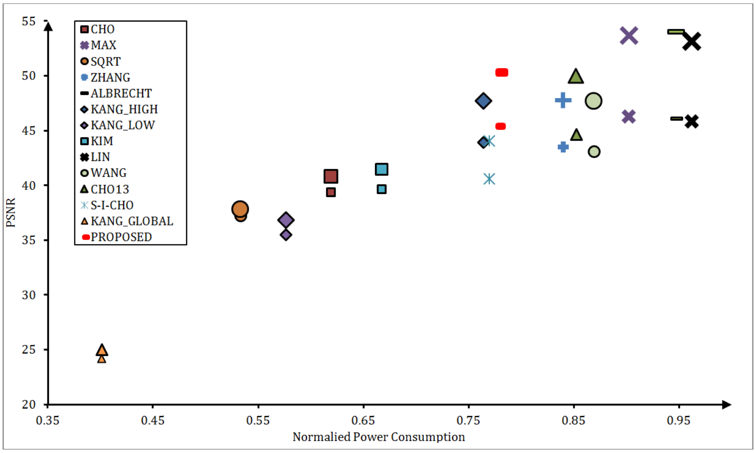


Figure 3.15: PSNR/power of the proposed algorithm using local histograms compared with other algorithms on edge-lit display. The smaller markers indicate $\varepsilon = 0.001$, the bigger ones $\varepsilon = 0.0002$ [4].

3.1.4.3 Subjective Evaluations

In order to assess the performance of the proposed algorithm and generally verify the validity of the backlight dimming LCD model, the visual quality of the results was tested in a subjective image quality experiment. 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 sat in front of the display at a distance equal to

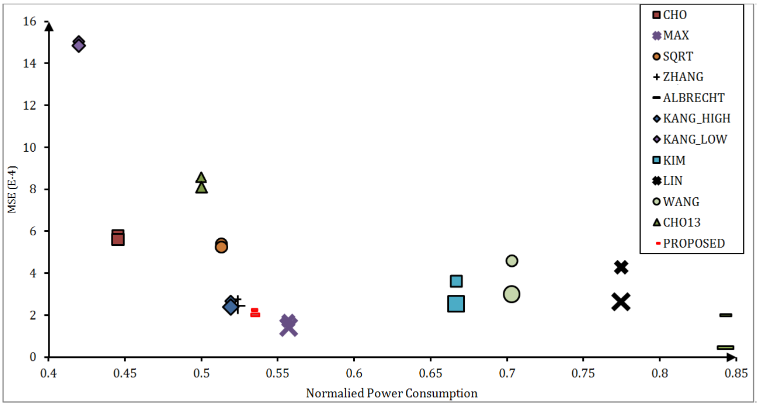


Figure 3.16: PSNR/power of the proposed algorithm using local histograms compared with other algorithms on direct-lit SIM2 display. The smaller markers indicate $\varepsilon = 0.001$, the bigger ones $\varepsilon = 0.0002$ [4].

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 were used, three among those were 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 subjective preferences were transformed into the absolute ratings using the Thurstone-Mosteller model described in [91], [92] and the maximum a posteriori model to deal with unanimous ratings (i.e. when all or almost all participants prefer one version to the other), as described in [93]. Table 3.1 shows the ratings scaled to the same range: the higher the grade is, the more participants preferred this version.

In most cases, the proposed method was ranked in the second place behind Albrecht [69], before Zhang [80] and was always in the first three places. Both Albrecht and Zhang however, have higher power consumption. This confirms the impressions of the objective measurements. The subjective quality ranking follows reasonably well the objective quality ranking based on PSNR, suggesting PSNR can be used to approximate the relative visual quality in simulated backlight dimming systems. Additional objective and objective results can be found in [4], [10].

Table 3.1: Subjective (Subj.) rankings of the proposed (Prop.) algorithm using local histograms compared with other algorithms [4].

Image	Grade Type	Prop.	Zhang [80]	Albrecht [69]	Kang Low [82]	Cho [78]
Stars	Subj.	0.56	0.58	0.58	0.52	0
	PSNR (dB)	35.40	33.76	33.76	35.83	35.69
	Power	0.59	0.98	0.98	0.50	0.49
Beach	Subj.	0.86	0.73	0.94	0	0.76
	PSNR	57.06	53.54	57.34	26.07	48.15
	Power	0.89	0.80	0.85	0.45	0.77
Parrot	Subj.	0.99	0.84	0.88	0	0.45
	PSNR	53.97	53.55	54.35	28.08	42.88
	Power	0.87	0.81	0.84	0.45	0.66
Volcano	Subj.	0.55	0.73	0.76	0.14	0
	PSNR	34.98	37.06	37.83	28.14	30.65
	Power	0.54	0.77	0.66	0.38	0.44
Exotic Flower	Subj.	0.27	0.52	0.85	0.18	0
	PSNR	37.61	41.88	45.42	34.84	28.26
	Power	0.81	0.93	0.98	0.75	0.63
Lizard	Subj.	0.67	0.50	0.87	0	0.05
	PSNR	40.19	45.13	45.15	32.72	33.04
	Power	0.75	0.91	0.97	0.55	0.59
Diver	Subj.	0.45	0.37	0.51	Excl.	0
	PSNR	44.93	37.02	45.85	26.08	30.65
	Power	0.70	0.74	0.82	0.47	0.56
Average	Subj. Rank	2.28	2.43	1.28	4.57	4.43
	PSNR	43.45	43.13	45.67	30.25	35.62
	PSNR Rank	2.57	2.71	1.57	4.29	3.86
	Power	0.74	0.85	0.87	0.51	0.59

3.2 Flicker Reduction in LED-LCDs with Local Backlight

Local backlight dimming of LCD with LED backlight can reduce the power consumption and improve the quality of displayed images. The images are displayed on the LCD screen with the temporal changes within the video sequences. Therefore, the display performance is characterized by both the spatial as well as the temporal images qualities. When the dimming algorithm is applied over time in video sequences, the important variations of LED over the time produce another visually annoying artifact called flickering. This section addresses flickering artifact in local LED-LCD backlight and techniques that can reduce it while maintaining the video quality. This work is based on [5].

Some algorithms were proposed for removing the flickering artifact in the global backlight dimming methods [89], [94], [95]. These algorithms can not be adapted with the local LED-LCD backlight dimming methods. Therefore, some algorithms were proposed for removing the flicker in the local backlight dimming methods [70], [96]. All of them use the Average algorithm [76] to remove the flicker effect. In Section 3.2.1, two State of the Art algorithms will be described. Then, a new algorithm will be presented for flicker reduction.

3.2.1 State of the Art Flicker Reduction Algorithms

As mentioned in Section 3.1.2, there are several algorithms for backlight dimming. Some algorithms use the characteristic data of the target image [4], [79], [80], while more complex algorithms, use the information of the PSF to model the light diffusion [15], [16], [69]. By using these algorithms, LCDs achieve high contrast ratio and low power consumption.

A rendered video frame is formed by multiplying the light emitted from the backlight with the transmittance of the LC pixels [4]. In video sequences, when the image content changes or moves very quickly and frequently, the calculated backlight luminance can also change very rapidly and frequently as shown in Fig. 3.17. This phenomenon creates an unnatural trembling in the backlight segment, which defines the flickering artifact and is caused by temporal local or global dimming.

Some algorithms for flicker removal were proposed in the literature [70], [96]. The first algorithm uses Moving Average (MA) filter taking the average pixel values through several consecutive frames. Since, this algorithm can be simply implemented, it is generally applied to the local dimming LCDs. However, the MA filtering can not be generalized to the other dimming algorithms like PSF-based or histogram-based algorithms, because it uses the average luminance value of the target image for removing flickering artifact. Most of the backlight dimming algorithms of today use other features (PSF or histogram) [4], [79], [80]. In addition, due to the MA calculation over a number of consecutive frames, it causes a delay in the backlight luminance when a change in a scene occurs.

This section presents briefly two State of the Art algorithms for removing flickering artifact in local LED-LCD backlight. These algorithms were used for comparison against the proposed algorithm.

3.2.1.1 Lee Algorithm

The MA filtering method can effectively reduce the flicker artifacts, but causes another problem known as backlight-response-time delaying. Hence, Lee et al. [96] presented an adaptive moving average filtering method to reduce the flickering artifact in the local backlight display. To find the appropriate backlight luminance, the method uses the changes on the average luminance of consecutive frames. This method can also detect movement existing in only a partial region of a frame that can not be found by conventional methods [96].

In the first step, the amount of the average luminance change ($R_x(t,k)$) of the x^{th} segment between the current frame (t^{th} frame and the k^{th} backward frame ($(t-k)^{th}$ frame) is calculated as follows:

$$R_x(t,k) = |avg_x(t) - avg_x(t-k)|, \tag{3.27}$$

Let define the tap-length of x^{th} block in t^{th} frame as $L_x(t)$. For some value $k = k_1$, if the $R_x(t,k)$ exceeds a pre-defined threshold T, there exists movement between the current frame and the k^{th} backward frame in the x^{th} segment. Therefore, the $L_x(t)$ should be set shorter than k_1 so that the backlight response time delaying can be avoided. Similarly, if the $R_x(t,k)$ does not exceed the predefined threshold T for all values which satisfies $k \leq k_2$, no effective movement exists between the current frame and the k^{th} backward frame in the x^{th} segment. In this case, the $L_x(t)$ should be larger than k_2 to reduce the flicker problem. This iterative process is performed for all segments in each frame whenever the tap-length is determined. After calculating the tap-length $(L_x(t))$, the backlight luminance is averaged based on the length of tap-length to reduce the flickering artifact. More information can be found in the original paper [96].

3.2.1.2 Chen Algorithm

In [70], an Infinite Impulse Response (IIR) filter was proposed to smooth the temporal backlight variation and eliminate the flickering. This algorithm uses the temporal and spatial backlight luminance to avoid flickering artifact [70]. The proposed filter is designed basically as follows:

$$L_{TF}^{t} = \sigma L_{init}^{t} + (1 - \sigma) L_{TF}^{t-1}, \tag{3.28}$$

where L_{TF}^{k} and L_{init}^{k} are the output of the temporal filter and initial backlight values, t and $t-1$ show the current and the previous frames, and σ is a controlling parameter of the temporal filter [70].

The MA and adaptive MA filtering can not be generalized to the other dimming algorithms like PSF-based or histogram-based algorithms, because they use the average luminance value of the target image for removing the flickering artifact. However, most of the backlight dimming algorithms use other features (PSF or histogram) [4], [79], [80]. In addition, they cause a delay in the backlight when a scene change occurs. Chen's algorithm [70] is a first order Infinite Impulse Response (IIR) filter in which, the filter coefficients are calculated using the average values of the target image. While most flicker removal methods are designed for a specific backlight algorithm, they can not be generalized to be used in the other algorithms. The average values of the target image are considered as a feature in most flicker removal algorithms. The proposed algorithm [5] is an adaptive flicker-reduction algorithm for LED-LCDs with dynamic backlight. The proposed algorithm is a second order IIR filter which uses the information of the two previous spatial and temporal frames to calculate the backlight luminance. The filter coefficients are calculated adaptively using the local features of the target image.

The difference between the maximum and the average luminance values of the target image in each backlight segment is used to calculate the filter coefficients.

3.2.2 Flicker Reduction Using IIR Filter

An annoying artifact that may be caused by the local dimming is screen flicker. When the frame content of a video sequence changes or moves very quickly and frequently, the calculated LED luminous intensities may also change likewise. Rapid backlight can cause flickering. When the backlight luminance intensity changes more than the pixel values do, the flicker artifact will be more obvious. Therefore, a temporal filter is required to reduce the flicker artifact by temporally smoothing the change in the backlight. Figure 3.17 shows an example of the flickering artifact on an edge-lit display when the Cho algorithm [79] is applied to a video sequence.

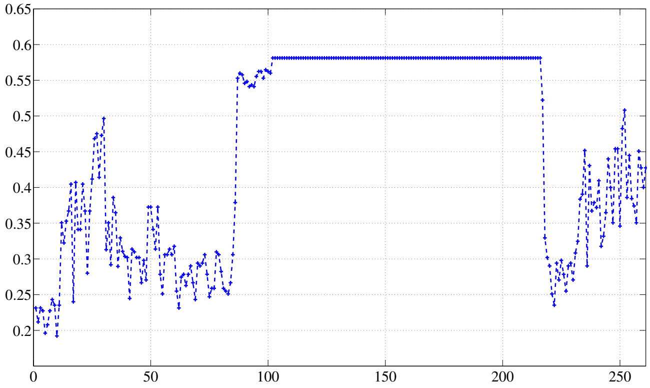


Figure 3.17: Example of the flickering artifact in LED segment No.4 of the edge-lit display (The video is *'Stars'* [97] on Cho's algorithm [79] with 261 frames) [5].

An IIR filter can reduce the flickering artifact. One problem with using a smooth IIR filter is that the change in the backlight lags behind the change of the image signal. This artifact is more obvious at the moment of scene change, especially from a bright scene to a black scene. To reduce and remove screen flicker and backlighting lag artifacts an adaptive second order IIR is proposed. Figure 3.18 illustrates the block diagram of the proposed IIR filter.

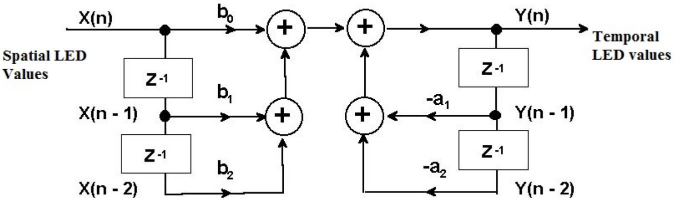


Figure 3.18: Block diagram of the proposed IIR for the flickering reduction. (Z^{-1} is the delay operator) [5].

The difference equation of the second order IIR filter in general is as follows:

$$y[n] = a_1 y[n-1] + a_2 y[n-2] + b_0 x[n] + b_1 x[n-1] + b_2 x[n-2] \quad (3.29)$$

where $y[n]$ and $x[n]$ are output and input of the filter. In order to calculate the temporal output of the backlight luminance, a linear combination of the spatial and the temporal variables is used as shown in Eq.3.29. The temporal backlight luminance (the filtered LED value) is as follows:

$$L_T^k = (a_1 L_T^{k-1} + a_2 L_T^{k-2} + b_0 L_S^k + b_1 L_S^{k-1} + b_2 L_S^{k-2})/\lambda \quad (3.30)$$

where L_T^k and L_S^k are the LED values after and before filtering that correspond to $y[n]$ and $x[n]$ of the IIR filter, k, $k-1$ and $k-2$ indicate the current and the last two frames, and a_1, a_2, b_0, b_1 and b_2 are the parameters controlling the smoothness of the IIR low-pass filter. The coefficients are selected adaptively according to the video content and are calculated as:

$$\begin{aligned}
&b_2 = 0;\\
&b_0 = \min(1, TH_1 + |diff_m^k - diff_m^{k-1}|);\\
&b_1 = \min(1, TH_2 + |diff_m^{k-1} - diff_m^{k-2}|);\\
&a_1 = 1 - b_0;\\
&a_2 = 1 - b_1;\\
&and\\
&\lambda = a_1 + a_2 + b_0 + b_1 + b_2;
\end{aligned} \quad (3.31)$$

where TH_1 and TH_2 are predefined parameters used to control the shape of IIR filter, k is the frame number, m is the backlight segment number, and $diff_m^k$ is the difference between the average and the maximum of the normalized pixel values in the entire segment. It is calculated as follows:

$$diff_m = max_m - avg_m \tag{3.32}$$

where max_m and avg_m are the maximum and the average values of the normalized pixel values in the physical domain. A larger difference between $diff^k$, $diff^{k-1}$ and $diff^{k-2}$ increases the possibility of scene change and therefore, larger coefficient values are required. This means that, the temporal output is not smooth enough. Using the adaptive IIR filter, the changes in the backlight output increases when there are changes in the scene, otherwise the backlight changes smoothly. In this way, both flicker and backlight lag are reduced.

3.2.2.1 Experiments

The performance of the proposed algorithm was tested on some State of the Art backlight dimming algorithms. The algorithms were Albrecht's algorithm [69], Cho's algorithm [79], Zhang's algorithm [80], Nadernejad's algorithm [4] and the optimization algorithm [16], [10]. For Albrecht's algorithm [69], we have implemented the first two steps (out of three, the third one was optional), which produces a clipper-free result. Two versions of the optimization algorithm were implemented [10]. These are based on Gradient Descent (GD) optimization algorithm with two leakage factors (GD with $\epsilon = 0.001$, and GD with $\epsilon = 0.0002$ [10]). All algorithms were simulated on an edge-lit display with 16 backlight segments placed in eight rows and two columns.

To assess the performance of our algorithm on video sequences, we tested them on 8 Full-HD (1920x1080) sequences. Four sequences (*Volcano*, *Anemone*, *BBBunny* and *Diver*) come from [98], two (*Stars* and *Titles*) come from [97] and two come from DVD content. These sequences present varying characteristics in terms of high and low luminance, color and details, as well as temporal variation (motion and scene change). Figure 3.19 shows a frame extracted from each of them. In the experiments $TH_1 = TH_2 = 0.125$.

Figure 3.20 shows the result of the proposed algorithm on different

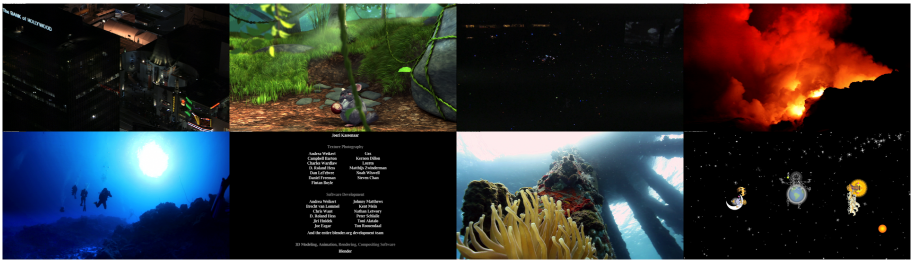


Figure 3.19: A snapshot of the 8 test video sequences from [98]. The name of sequences from left to right and top to bottom are: *Theatre*, *BBBunny* [98], *Concert*, *Volcano* [98], *Diver* [98], *Titles* [97], *Anemone* [98], and *Stars* [97], respectively [5].

video sequences for segment number 13 of edge-lit display using Cho's algorithm [79]. As can be seen, when the level of flicker is low like the *Volcano* and *Concert* sequences, the algorithm preserves the LED values. However, in the case of high flicker like *Stars* and *Theatre*, the fluctuations in LED values are controlled in an acceptable level. The control level follows the normal trend of the previous LED values. The effect of the proposed algorithm is the same for other segments. In Fig. 3.21, the average LED values (the average for all 16 segments of the edge-lit display) for one video sequence '*Stars*' are shown. Another advantage of this method is that, the average power consumption does not change significantly after flicker removal. The power level before and after filtering are shown at the bottom of Fig. 3.21.

In order to evaluate the performance of the proposed algorithm based on the objective quality measures, the PSNR and MSE vectors for *Stars* video sequence after and before filtering are illustrated in Figs. 3.22 and 3.23 (The LC values are quantized to 10 bits). As shown in the these figures, the proposed algorithm can produce a flicker-free result while keeping the quality of the video signal. Although, the PSNR and MSE after filtering were not always better compared to those of the original video sequence, their averages were almost the same as shown in the figures. In addition, their overall fluctuations were smoothed.

The average results for different local dimming algorithms on different sequences are summarized in Table 3.2. These results are for the *Title* (251 frames), *Theatre* (207 frames), *Stars* (255 frames), *Diver* (300 frames), *Concert* (217 frames), *Anemone* (300 frames), *Volcano*

(300 frames), and *BBBunny* (250 frames) video sequences. It is important to keep the quality of the videos at their initial level before filtering. Therefore, in this table, the minimum as well as the average PSNR are reported and due to its reverse relationship with the MSE, the corresponding maximum and average MSEs are shown. For power consumption, all the minimum, maximum and average values are reported. As observed in the table, the proposed filter does not have a large impact on PSNR, MSE and the power consumption, in terms of minimum, maximum and average values. In addition, figure 3.24 shows the average quality versus power consumption of the different algorithms applied on all the sequences before and after filtering. In general, a flicker removal algorithm should not affect the quality and power consumption of LEDs. These results verify that the proposed algorithm preserves these two characteristics since the position of the points in the plot are close to each other before and after filtering.

There is no objective metric for the flickering artifact in LCD displays with local backlight. However, we evaluated the performance of the filter both visually and objectively. The temporal variation is computed by the SSD as recommended in [42] to measure flickering caused by video compression. SSD is defined as:

$$SSD = \frac{1}{N_{frames}} \sum_{n} \frac{1}{N_{pixels}} \sum_{i} (Fr_{n-1}(i) - Fr_{n}(i))^2 \qquad (3.33)$$

where $Fr_n(i)$ is the pixel i of the frame n. The SSD results for all sequences are shown in Table 3.3. The average level of temporal variations is either smaller after temporal filtering (TF) than before applying the filter ($Orgn$) or similar. In some cases that the level of the flicker was quite high, the algorithm was applied two or three times for a successful removal. This is shown by the numbers inside the parentheses in Table 3.3. In some cases, the removal was not completely successful and this is shown by $(*)$ in the table that means not removed completely.

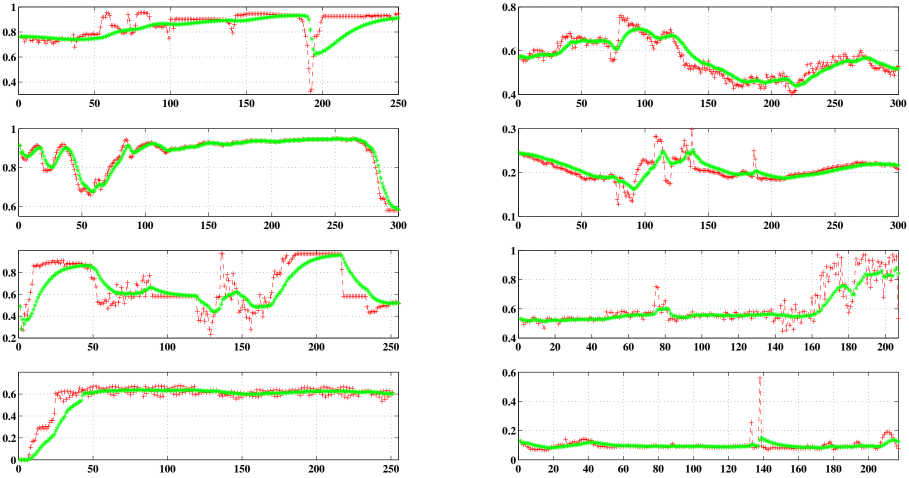


Figure 3.20: Performance of the proposed algorithm on different sequences, visualized using an edge-lit display in which the Cho's algorithm is used [5]. The red and green line show the LED values before and after filtering, respectively [5]. LED No.13, Column. 1: 1,...4 are: *BBBunny*, *Volcano*, *Stars*, *Title*, and column. 2: 1,...4 are: *Anemone*, *Diver*, *Theatre*, *Concert*. The horizontal and vertical axis are the frame number and LED values, respectively.

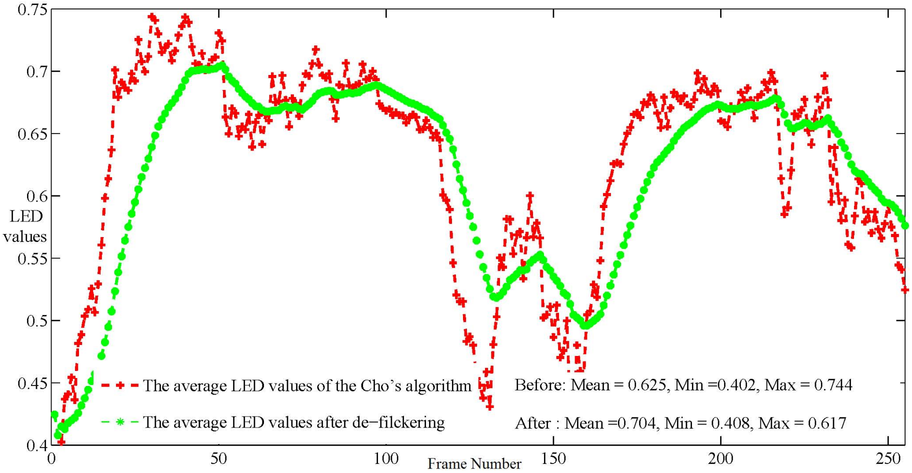


Figure 3.21: The average led values for 16 backlight segments before and after filtering applied on *Stars* using Cho's algorithm [5].

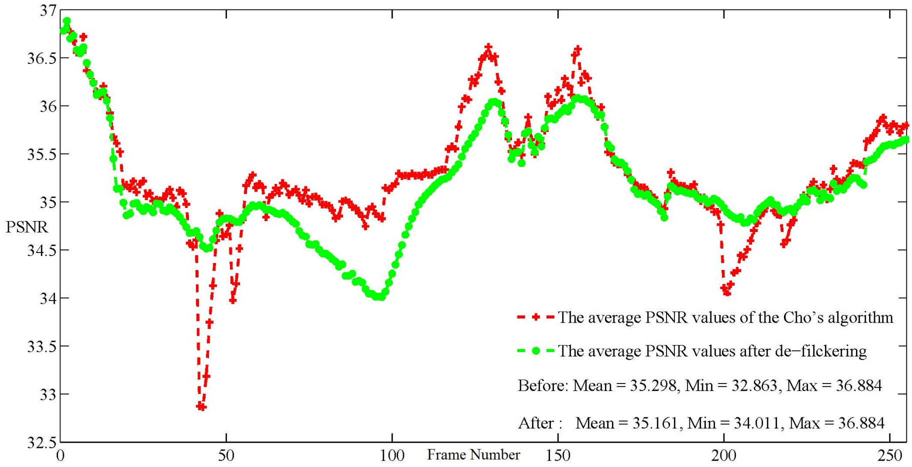


Figure 3.22: PSNR of the *Stars* sequences before and after de-flickering [5].

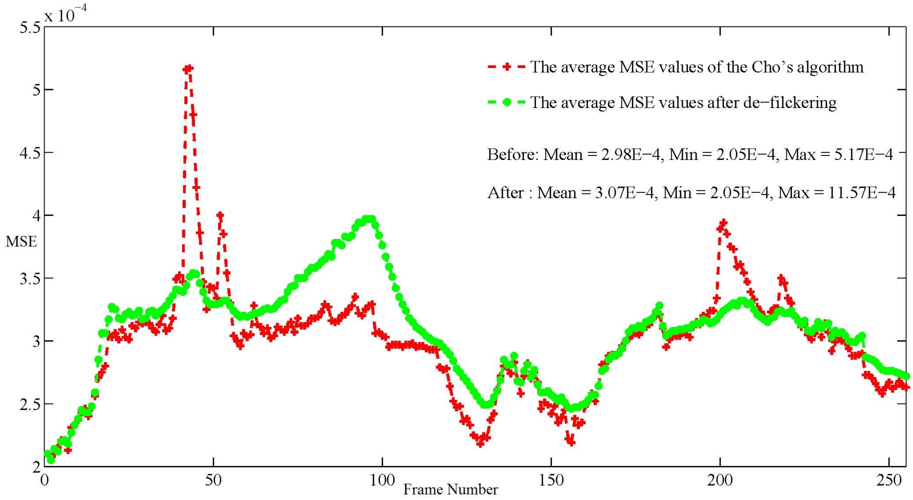


Figure 3.23: MSE of the *Stars2* sequences before and after de-flickering [5].

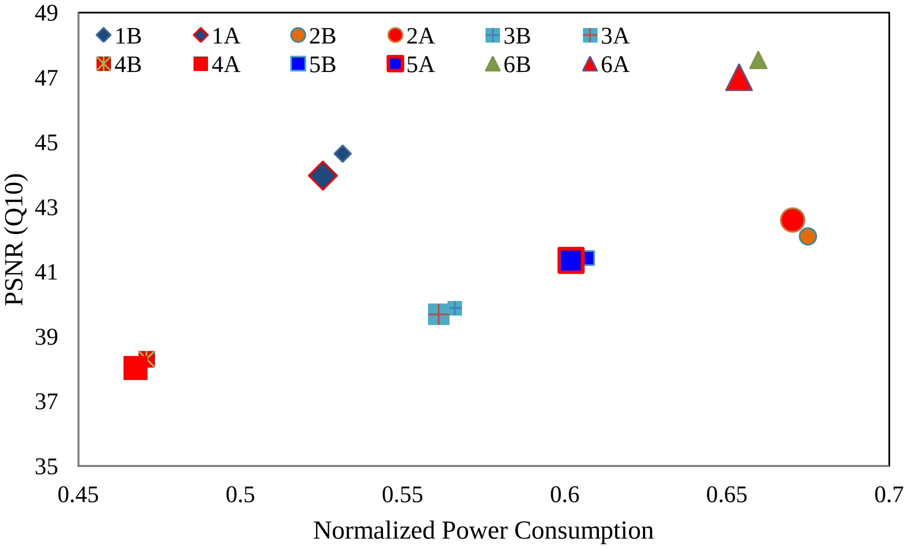


Figure 3.24: The average PSNR vs. power consumption for different sequences before and after filtering [5]. The numbers are: 1: GD with $\epsilon = 0.0002$ [10], 2: Albrecht [69], 3: Cho [79], 4: Nadernejad [4], 5: Zhang [80], and 6: GD with $\epsilon = 0.001$ [10], respectively. A and B show before and after filtering results.

Table 3.2: The average de-flickering results for different algorithms on all video sequences [5]. The algorithms are: 1: GD with $\epsilon = 0.0002$ [10], 2: Albrecht [69], 3: Cho [79], 4: Nadernejad [4], 5: Zhang [80], and 6: GD with $\epsilon = 0.001$ [10], respectively.

		Before					
Metrics ⇓	Algorithms ⇒	1	2	3	4	5	6
MSE(E-4)	Mean Max	0.824 1.130	2.737 5.787	2.385 4.093	3.769 9.316	1.765 4.877	1.108 1.526
PSNR	Mean Min	47.537 44.898	42.090 37.354	39.878 37.393	38.304 35.129	41.419 38.150	45.092 42.778
Power	Mean Min Max	0.660 0.480 0.782	0.675 0.492 0.828	0.566 0.433 0.673	0.471 0.377 0.548	0.607 0.470 0.710	0.531 0.377 0.646
		After					
	Algorithms ⇒	1	2	3	4	5	6
MSE(E-4)	Mean Max	0.957 1.455	2.639 5.078	2.410 3.935	3.927 7.530	1.769 3.385	1.522 2.641
PSNR	Mean Min	47.005 43.458	42.595 37.689	39.680 36.108	38.024 34.580	41.349 37.544	44.285 39.927
Power	Mean Min Max	0.668 0.538 0.724	0.697 0.597 0.653	0.572 0.479 0.596	0.476 0.409 0.495	0.613 0.520 0.636	0.525 0.401 0.624

Table 3.3: SSD measures of sequences before and after temporal filtering for different algorithms on all video sequences [5]. (The numbers inside the parentheses show the number of times the filtering was repeated and (∗) means that the flicker was not removed completely.)

		GD $\epsilon = 0.0002$ [10]		Albrecht [69]		Cho [79]	
		Orgn	TF	Orgn	TF	Orgn	TF
Stars	Avg	64,83	59,79	67,34	66,43	65,37	64,04
	Max	137,87	122,78	140,73	138,74	133,88	134,24
Concert	Avg	0,71	0,70 (2)	0,63	0,56 (2∗)	0,68	0,65
	Max	6,82	6,09	7,00	3,28	6,65	4,04
Titles	Avg	117,94	115,77	72,13	71,60	93,15	92,46
	Max	231,10	229,15	145,09	139,69	181,30	179,56
Volcano	Avg	1,16	1,16	1,18	1,17 (2)	1,10	1,10
	Max	1,88	1,88	1,90	1,88	1,76	1,76
Anemone	Avg	27,99	27,98	27,84	27,88 (2)	27,93	27,92
	Max	78,29	78,28	78,24	78,28	78,16	78,15
BBBunny	Avg	72,56	72,39 (2)	72,39	72,34 (2)	71,25	70,90 (2)
	Max	288,17	287,28	287,61	287,04	283,84	282,65 6
Theatre	Avg	32,75	32,71 (2∗)	32,92	32,91 (3∗)	30,84	30,89
	Max	47,54	47,58	48,05	47,76	45,79	45,57
Diver	Avg	4,45	4,44 (2)	4,46	4,45 (2∗)	4,27	4,26
	Max	12,30	12,25	12,34	12,35	11,85	11,83
		Nadernejad [4]		Zhang [80]		GD $\epsilon = 0.001$ [10]	
		Orgn	TF	Orgn	TF	Orgn	TF
Stars	Avg	55,79	52,87	65,24	63,72	59,66	52,37
	Max	115,24	112,01	135,55	135,17	131,10	110,54
Concert	Avg	0,66	0,63 (2∗)	0,69	0,66	0,69	0,66 (2
	Max	5,85	3,59	6,54	4,45	6,43	4,60
Titles	Avg	70,48	69,62	92,54	91,85	108,09	103,09
	Max	139,82	136,54	183,50	179,03	213,42	207,80
Volcano	Avg	1,02	1,02	1,14	1,14	1,12	1,12
	Max	1,65	1,64	1,85	1,85	1,82	1,82
Anemone	Avg	27,79	27,79	27,95	27,94	27,91	27,91
	Max	78,01	78,04	78,22	78,20	78,23	78,28
BBBunny	Avg	70,93	70,81	72,01	71,86	72,37	72,02 (2)
	Max	283,71	282,06	286,77	285,28	287,92	285,76
Theatre	Avg	28,04	28,09	31,90	31,91	31,35	31,36
	Max	41,81	41,81	46,51	46,32	45,34	45,49
Diver	Avg	4,32	4,31	4,33	4,32	4,42	4,41 (2
	Max	11,88	11,88	12,07	12,07	12,20	12,17

3.3 Conclusion

The reduction of power consumption is one of todays most important topics in the LCD industry. The backlight is the main consumer of power in LCDs. In conventional LED-LCD backlight, the backlight acts as a constant light source and consumes the same power for dark as well as for bright images. By using local dimming algorithm, the power consumption can be reduced while the contrast ratio is increased. When the dimming algorithm is applied over time in video sequences, the important variations of LED over time produce another visually annoying artifact, called flickering.

This Chapter presented two adaptive algorithms for local backlight dimming and flicker artifact reduction. The dimming algorithm is a low complex and adaptive algorithm for local dimming of LED backlight displays which uses local characteristics of the target image, such as the local histograms and the average pixel intensity, to find the backlight luminance. Results show that the proposed algorithm can achieve better trade-off between power consumption and image quality. In addition, experimental results of de-flickering demonstrated that the algorithm can avoid abrupt variations of the LED signals and consequently reduce the flicker artifact.

Chapter 4

Conclusion and Future Work

Today, image and video signals have widespread use in our life. A huge range of our information is saved as images or video signal. For saving and transmission of this information, compression is required. After compression, the visual quality of the compressed information is not the same as the original one. The compressed image and video signal have artifacts. Therefore, they should be processed to improve the visual quality. The visual quality of the compressed video signal can be enhanced using post-processing techniques.

The visual quality is important for perceived image and video signals, particularly for TV screens. The new LCDs allow to save the power and improve quality, dynamically. They have high contrast ratio and can also improve the image quality. A typical Liquid Crystal Display (LCD) consists of a backlight located behind a matrix of pixels, made of liquid crystal (LC) cells that are basically voltage controlled light filters. In order to perform both the power saving as well as quality improving, dimming algorithms are used.

This thesis presented several techniques to improve image quality through the use of dynamic local backlights or post-processing.

First, three novel algorithms for blocking and ringing removal in images coded with Joint Photographic Experts Group (JPEG) and sequences coded with MPEG-2 or H.264/AVC were presented. The three approaches exploit the joint benefits of anisotropic diffusion and fuzzy filtering (in one case), bilateral and trilateral filters (in the others).

In the first algorithm [2], a method combining the powerful anisotropic diffusion equations with fuzzy filtering for removing blocking and ringing artifacts was presented. Due to the directional nature of these artifacts, directional anisotropic diffusion was applied. In order to improve the performance of the algorithm, a threshold parameter for the diffusion coefficient was selected adaptively. Two different methods based on this approach were presented, one was designed for still images and the other for YUV video sequences. For the video sequences, different filters were applied to luminance (Y) and chrominance (U,V) components. The performance of the proposed method was 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, indicated that the proposed algorithms employing directional and spatial fuzzy filters performed better for artifact reduction than other methods.

In the second algorithm [3], a method for reducing the blocking and ringing artifacts in H.264/AVC video sequences was presented. For deblocking, the proposed method used a quality measure of a block based coded image to find filtering modes. Based on filtering modes, the images were segmented into three classes and a specific deblocking filter was applied to each class. Deringing was performed by an adaptive bilateral filter; the spatial and intensity spread parameters were selected adaptively using texture and edge mapping. The analysis of objective and subjective experimental results showed that the proposed algorithm was effective in deblocking and deringing of low bit-rate H.264 video sequences.

The third algorithm employed an adaptive trilateral filter for removing the artifacts. It is non-iterative and its complexity is low. In addition, it does not require motion estimation and compensation steps for post-processing. In the proposed algorithm, an extra spatial Gaussian multiplicative filter was added to the bilateral terms for removing spatial and temporal artifacts. To evaluate the effectiveness of the proposed algorithm in the temporal dimension, a temporal measurement was also considered for all test sequences. Comparison of the experimental results shows that the proposed algorithm effectively reduces flickering artifacts and simultaneously removes blocking and ringing. The third algorithm provided best performance in terms of objective and subjective tests comparing to the other existing algorithms.

The State of the Art post-processing and backlight dimming algorithms were implemented. In addition, a new algorithm for backlight dimming was proposed [4] that exploited the characteristics of the target image, such as the local histograms and the average pixel intensity of each backlight segment, to reduce the power consumption of the backlight and enhance the image quality. The results showed that the algorithm can provide a better trade-off between the power consumption and image quality preservation than the other feature based dimming algorithms.

Finally, a novel flicker reduction algorithm in LED-LCDs with local backlight was presented. The proposed algorithm used an adaptive second order Infinite Impulse Response (IIR) filter in which coefficients were calculated from the local image features. Experimental results showed that the proposed method can reduce flickering while simultaneously keeping similar video quality in terms of PSNR, MSE and SSD.

The backlight dimming algorithms and post-processing algorithms were used independently, while they were dependent to each other. For future work, the main goal can be to develop an intelligent algorithm capable to perform the three main tasks: artifact reduction of the decoded video (de-blocking and de-ringing), minimization of the power consumption in LED-LCD backlight and improvement of the video quality (increasing the contrast ratio and reducing the flickering) in a unified algorithm. In addition, the backlight dimming algorithms can be used for processing of HDR images to be visualized on Ultra HD displays. Then, an optimal algorithm should be developed to improve the visual quality on the HDR-Ultra HD display. For this aim, the optimal tone mapping and dimming algorithms can be integrated to find the optimal algorithm.

Appendices

Appendix A

Ph. D. Publications

A.1 Artifact Reduction of Compressed Images and Video Combining Adaptive Fuzzy Filtering and directional Anisotropic Diffusion

E. Nadernejad, S. Forchhammer, and J. Korhonen, “Artifact Reduction of Compressed Images and Video Combining Adaptive Fuzzy Filtering and Directional Anisotropic Diffusion”, in *2011 3rd European Workshop on Visual Information Processing (EUVIP)*, IEEE, 2011, pp. 24–29

Reference: [1]

ARTIFACT REDUCTION OF COMPRESSED IMAGES AND VIDEO COMBINING ADAPTIVE FUZZY FILTERING AND DIRECTIONAL ANISOTROPIC DIFFUSION

Ehsan Nadernejad, Søren Forchhammer, and Jari Korhonen
Dept. of Photonics Engineering, Technical Univ. of Denmark, Kgs. Lyngby, Denmark

ABSTRACT

Fuzzy filtering is one of the recently developed methods for reducing distortion in compressed images and video. In this paper, we combine the powerful anisotropic diffusion equations with fuzzy filtering in order to reduce the impact of artifacts. Based on the directional nature of the blocking and ringing artifacts, we have applied directional anisotropic diffusion. Besides that, the selection of the adaptive threshold parameter for the diffusion coefficient has also improved the performance of the algorithm. Experimental results on JPEG compressed images as well as MJPEG and H.264 compressed videos show improvement in artifact reduction of the proposed algorithm over other directional and spatial fuzzy filters.

Index Terms— Artifact Reduction, Anisotropic Diffusion, Fuzzy filter, H.264.

1. INTRODUCTION

Image and video compression suffers from spatial and temporal distortions. Spatial distortion includes blocking and ringing, and 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. It is also known as Gibbs Phenomenon. When ringing artifacts are changing from frame to frame as a video sequence is displayed, mosquito artifacts are created. Flickering artifacts [7] appear due to the inconsistency in quality at the same spatial position in adjacent frames.

Over the years, many algorithms have been proposed to reduce the spatial and temporal artifacts. For de-blocking, linear low-pass filters in [1,2] and Projection Onto Convex Sets “POCS” [3] have been used, but these methods cause blurriness and have a high computational complexity. To remove ringing artifacts, linear or non-linear isotropic filters can be applied to the regions near to edges [4, 5]. For combating flickering artifacts, most of the current methods focus on reducing the flickering in intra-frame coding [7,8]. In [7], the quantization error is considered when obtaining the optimal intra prediction mode and to help reducing the flickering. Also in [8], flickering is included in the cost function, when the optimal prediction mode and block-size are chosen. In [9,10], spatiotemporal fuzzy filters are used to remove different artifacts.

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 [11,16].

Fuzzy filters are improved median filters or a rank condition rank selection filters [14], where the binary spatial-rank relation is replaced by a real valued relation. This permits the filter to adapt to the spread of the signal by averaging the flat areas, while the isolated pixels in the edge areas remain.

In this paper, we propose a new algorithm using anisotropic diffusion processing and spatial fuzzy filtering [9,10] to reduce coding artifacts in compressed images and video. We process all the vertical and horizontal artifacts using one-dimensional (1D) anisotropic diffusion after applying the fuzzy filter. To avoid the blurring effect, anisotropic diffusion is performed with a small number of iterations.

The rest of the paper is organized as follows: Section 2 summarizes the anisotropic diffusion. Section 3 provides a description of fuzzy filtering. Adaptive fuzzy filtering and anisotropic diffusion for reducing artifacts in compressed video sequences are combined in Section 4. Section 5 shows the simulation results and compares the proposed algorithm with known methods. Finally, the concluding remarks are given in Section 6.

2. ANISOTROPIC DIFFUSION

Diffusion is a fundamental physical process. The isotropic diffusion process can be modeled as a Gaussian smoothing process, where its variance increases continuously. For the anisotropic diffusion, the smoothing process may be performed differently in each direction. Let $I(x,y,t)$ represent an image at the coordinates (x, y) at time t of the diffusion, when the diffusion flux is defined as:

The work presented was in part funded by the Danish Strategic Research Council (DSF Nr. 09-067034).

978-1-4577-0071-2/11/$26.00 ©2011 IEEE

$$\varphi = -C\ \nabla I \qquad (1)$$

where C is the diffusion coefficient. With the matter continuity equation we get:

$$\frac{\partial I}{\partial t} = -\nabla \bullet \varphi \qquad (2)$$

By combining (1) and (2), the diffusion equation is obtained:

$$\frac{\partial I}{\partial t} = \nabla \bullet (C \nabla I) \qquad (3)$$

where "**.**" represents the inner product of two vectors. When C is a constant parameter, the diffusion process is isotropic. When C is a function of the directional parameters, the diffusion process becomes anisotropic. To solve the partial differential equation (3), the image I_0 is used as the initial condition and the Neumann boundary condition is applied to the image borders:

$$I(x,y,t)_{t=0} = I_0, \quad \partial_n I = 0 \qquad (4)$$

The discrete form of the diffusion equation is given by:

$$\frac{\partial I(x,y,t)}{\partial t} = \nabla \bullet (C(x,y,t) \nabla I(x,y,t)) \qquad (5)$$
$$I(x,y,0) = I_0(x,y)$$

where C is the diffusion coefficient and I is the original image in each time scale t.

Perona-Malik (*PM*) [11] suggested the two well-known diffusion coefficients (6) and (7):

$$C_1(s) = \frac{1}{1+\left(s/k\right)^2} \quad (6) \qquad C_2(s) = \exp\left[-\left(\frac{s}{k}\right)^2\right] \quad (7)$$

where $s = |\nabla I|$. By using diffusion equations, the diffusion process will be prominent when the magnitude of the local gradient is low, and will be restricted when the magnitude of the local gradient is high.

In both (6) and (7), parameter k is a factor controlling the diffusion process. Both equations provide perceptually similar results, but (6) emphasizes noise removal, while (7) emphasizes high contrast preservation.

In some cases, the *PM* method can misinterpret noise as edges and create false edges by enhancing the noise. Therefore, Catte *et al* [12] changed $s = |\nabla I|$ in the *PM* diffusion function to:

$$s = |\nabla G_\sigma * I| \qquad (8)$$

Here, G_σ is a Gaussian smoothing kernel and "$*$" is the convolution operator. In this approach, $|\nabla G_\sigma * I|$ is used to estimate the local gradient instead of the more noise-sensitive $|\nabla I|$.

A new method is introduced to find parameter k in diffusion coefficient (6,7). The idea is based on calculating the gradient of the image in different directions. The experiments on different images show that this parameter depends on the changes of the image gradient in different directions,

$$k = \alpha \cdot v \qquad (9)$$

where, α is a constant and v is the variance of the image gradient in different directions. By choosing $\alpha = 2 \cdot 10^{-6}$ the optimal results were obtained. The parameter was calculated based on experiments on 100 standard images from [17].

3. FUZZY FILTER

Fuzzy filters improve the median filters or rank condition rank selection filters [10, 13] by replacing the binary spatial-rank relation by a real-valued relation. In [14], fuzzy filter is defined by generalizing the binary spatial-rank relation. Assuming that filter h is applied to a set ω of neighboring samples $f(i+i',j+j')$ around the input $f(i,j)$, we can formulate the output:

$$g[i,j] = \sum_{[i',j']} h(f[i+i',j+j'], f[i,j]) \times f[i+i',j+j'] \qquad (10)$$

And its unbiased form via normalization:

$$g[i,j] = \frac{\sum_{[i',j'] \in \omega} h(f[i+i',j+j'], f[i,j])\, f[i+i',j+j']}{\sum_{[i',j'] \in \omega} h(f[i+i',j+j'], f[i,j])}, \qquad (11)$$

where $h(f[i+i',j+j'],f[i,j])$ controls the contribution of the input $f[i+i',j+j']$ to the output.

Due to the input independence of the filter coefficients, a low-pass filter designed to perform effectively in the flat areas may introduce blurring artifacts in the detailed areas. However, it is desirable to preserve the details, while removing the artifacts. This can be achieved by imposing constraints, such as if $f[i+i',j+j']$ is far from $f[i,j]$, its contribution to the output is small. In this case, the filter coefficients $h[i,j]$ must follow the constraints:

$$\lim_{|f[i+i',j+j']-f[i,j]| \to 0} h(f[i+i',j+j'], f[i,j]) = 1 \qquad (13)$$

$$\lim_{|f[i+i',j+j']-f[i,j]| \to \infty} h(f[i+i',j+j'], f[i,j]) = 0 \qquad (14)$$

$$\begin{aligned} &h(f[i+i_1',j+j_1'], f[i,j]) \ge h(f[i+i_2',j+j_2'], f[i,j]) \\ &if\ \left|f[i+i_1',j+j_1'] - f[i,j]\right| \le \left|f[i+i_2',j+j_2'] - f[i,j]\right|, \end{aligned} \qquad (15)$$

The function $h(f[i+i',j+j'],f[i,j])$ is called a membership function, and there are many functions that fulfill these requirements. A Gaussian membership function is given in:

$$h(f[i+i',j+j'], f[i,j]) = \exp(-\frac{(f[i+i',j+j'] - f[i,j])^2}{2\sigma^2}), \qquad (16)$$

where σ represents the spread parameter of the input and controls the strength of the fuzzy filter. The input $x[i,j]$ contributes the output always more that the other samples:

$$h(f[i,j], f[i,j]) \ge h(f[i+i',j+j'], f[i,j])\ \forall k \qquad (17)$$

For the same $|f[i+i',j+j']-f[i,j]|$, the higher the value of σ, the higher the contribution of $f[i+i',j+j']$ to the output. This implies that $f[i,j]$ will converge more towards $f[i+i',j+j']$. Smaller values of σ will keep the signal $f[i,j]$ 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 fixed spread parameters for every surrounding sample, ignoring their relative positions. In image and video compression, distortions such as blocking, ringing or flickering artifacts are directional, and, thus, the direction between *f*[*i,j*] and its surrounding samples *f*[*i+i',j+j'*] should be taken into consideration. This can be achieved by an adaptive spread parameter:

$$\sigma(f[i+i',j+j'],f[i,j]) = K[i,i',j,j'] \times \sigma_A[i,j] \tag{18}$$

where σ_A is a position-dependent amplitude function of the spread parameter, and *K* is the scaling function controlled by the direction of [*i+i',j+j'*] to [*i,j*]. In this paper, we use cosine-based general form for the spread parameter (19):

$$\sigma(\theta) = \sigma_A(\alpha + \beta\cos^2(\theta)), \tag{19}$$

where σ_A is a constant, θ is the direction between the pixel positions [*i,j*] and [*i+i',j+j'*], α and β are positive scaling factors controlling the maximum and minimum strength of the filter. The extensions of the membership function σ in (18) are discussed in [9,10] for compressed images and compressed video sequences. More information about fuzzy filters is available in [5,13,14].

4. PROPOSED METHOD FOR REMOVING ARTIFACTS

In the proposed method, adaptive 1-D fuzzy filtering [10] is first applied to the pixels suffering from blocking artifacts, and then directional anisotropic diffusion is used to increase the quality. Finally, adaptive 2-D fuzzy filtering [9] is applied to the pixels with ringing artifacts (see Fig. 1).

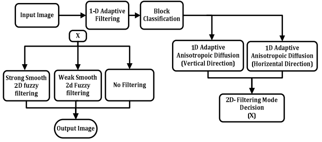

Input Image
1-D Adaptive Filtering
Block Classification
X
1D Adaptive Anisotropoic Diffusion (Vertical Direction)
1D Adaptive Anisotropoic Diffusion (Horizental Direction)
Strong Smooth 2D fuzzy filtering
Weak Smooth 2d Fuzzy filtering
No Filtering
2D- Filtering Mode Decision (X)
Output Image

Fig. 1 Flowchart of the proposed method.

In the first phase, vertical artifact detection is performed along each vertical boundary of an 8x8 block. For this purpose, the difference between each pair of boundary pixels is computed (see Fig. 2), and if MAX(*L1,L2,L3,L4*) < *G0*, or MAX(*R1,R2,R3,R4*) < *G0*, the current row is marked as a boundary gap and is filtered by a *1D* fuzzy filter. Similar analysis is performed also horizontally.

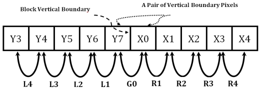


Fig. 2 Detect the vertical boundary gap in a row across the block vertical boundary.

At the block classification step, *2D* fuzzy filter is adapted to the variation of the pixels in each block. This is based on standard deviation (*STD*) of blocks (See Fig. 3).

$$STD(I(t,m,n)) = \sqrt{\frac{1}{9}\sum_{k=m-1}^{m+1}\sum_{l=n-1}^{n+1}\left(I(t,k,l)-\frac{1}{9}\sum_{p=m-1}^{m+1}\sum_{q=n-1}^{n+1}I(t,p,q)\right)^2} \tag{20}$$

The classification is based on the value of the maximum *STD* (*MaxSTD*) in an 8x8 block, as shown in (21).

$$MaxSTD\begin{cases}\in[45,+\infty) \Rightarrow & StrongEdge\\ \in[25,45) \Rightarrow & WeakEdge\\ \in[15,25) \Rightarrow & StrongTexture\\ \in[5,15) \Rightarrow & WaekTexture\\ \in[0,5) \Rightarrow & Smooth\end{cases} \tag{21}$$

where *MaxSTD* is the maximum *STD* of the pixels in each block.

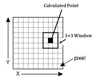


Fig. 3 Calculating Standard Deviation

To avoid smoothing and increase the quality of the images, *1D* anisotropic diffusion is performed instead of two-dimensional diffusion. Also, as mentioned in Section 2, the proposed anisotropic diffusion is an adaptive algorithm.

The numerical solution of the proposed algorithm (Adaptive Anisotropic Diffusion) is described below.

1- Find the value of *k* parameter based on (9).

2- Let the time step be Δt and the spatial step be *h* in *x, y* directions. Then, the time and space coordinates can be presented in discrete form as:

$$t = n\Delta, n = 0,1,2,....; x = ih, y = jh, \\ i = 1,2,3,...M-1, j = 0,1,2,...N-1 \tag{22}$$

where $Mh \times Nh$ is the size of the image. Let $I_{i,j}^n = I(ih, jh, n\Delta t)$ then the final image can be obtained using the four-stage approach described below:

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

$$\nabla^2 I_{i,j}^n(hor) = (I_{(i+1,j)}^n + I_{(i-1,j)}^n - 2\times I_{(i,j)}^n) \tag{23}$$

The symmetric boundary conditions are used:

$$I_{-1,j}^n = I_{0,j}^n, I_{M,j}^n = I_{M-1,j}^n, j = 0,1,2,....N-1, \tag{24}$$

2- Stage II: Computing the directional diffusion coefficient *c*(*x,y;t*), as for example:

$$c_{i,j}^n(hor) = \exp(-[\frac{\nabla^2 I_{i,j}^n(hor)}{2k^2}] \tag{25}$$

The other diffusion coefficients can be obtained in a similar fashion.

3- Stage III: Computing the divergence of $c(.)\nabla I$:

$$d_{i,j}^n = \frac{1}{h^2}[c_{i+1,j}^n(I_{i+1,y}^n - I_{x,y}^n) + c_{i-1,j}^n(I_{i-1,y}^n - I_{x,y}^n)], \tag{26}$$

With the symmetric boundary conditions:

$$d_{-1,j}^n = d_{0,j}^n, d_{M,j}^n = d_{M-1,j}^n, j = 0,1,2,....N-1, \tag{27}$$

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

$$I_{i,j}^{n+1}(hor) = I_{i,j}^n(hor) + \frac{\Delta t}{16} d_{i,j}^n(hor), \tag{28}$$

This equation is equivalent to (5) (in this work: Δ*t*/16≈0.125). The same algorithm is used in the vertical direction.

At the following step, 2D fuzzy filter is applied to each block based on *MaxSTD*. Figure 4 displays the flowchart of the criteria for selection of fuzzy parameters.

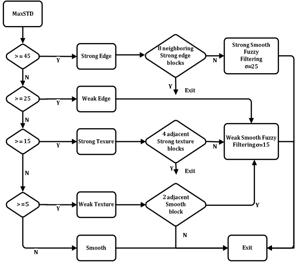

MaxSTD
>=45
Y
Strong Edge
8 neighboring Strong edge blocks
N
Strong Smooth Fuzzy Filtering σ=25
N
Y
Exit
>=25
Y
Weak Edge
N
>=15
Y
Strong Texure
4 adjacent Strong texture blocks
N
Weak Smooth Fuzzy Filtering σ=15
Y
Exit
N
>=5
Y
Weak Texture
2 adjacent Smooth block
Y
N
N
Smooth
Exit

Fig. 4 Selection of adaptive spread parameter σ.

In this algorithm (Fig. 1), the fuzzy filter is used to reduce blocking and ringing artifacts. The fuzzy filter helps reducing the artifacts while retaining the sharpness of the edges. The main drawback of this fuzzy filter is that it is isotropic for multi-dimensional signals while artifacts are directional. To avoid blurring, directional adaptive anisotropic diffusion is applied using few iterations.

To remove artifacts in video coded sequences H.264, a new algorithm is proposed (see Fig. 5). In this algorithm, first, the chroma components are up-sampled to the same size of the luma components. To obtain higher quality, each Y frame is enhanced by a directional spatial fuzzy filter and each U and V frame is de-noised in the same way as the algorithm in Fig.1.

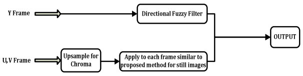


Fig. 5. Proposed method for video sequence

5. SIMULATION RESULTS

We tested the processing of MJPEG and H.264 decoded images of test video sequences. We have used two well known metrics to evaluate the algorithm. Peak Signal to Noise Ratio (PSNR) and Mean Structural Similarity (MSSIM) index are used for images, and Weighted-*PSNR* and Weighted-*MSSIM* [15] are used for the video coded sequences (i.e. H.264).

5.1 Enhancement of Compressed Images

Simulations have been performed to demonstrate performance of the proposed method. Different approaches are compared in terms of visual quality: *PSNR* and *MSSIM*. For the comparison, the methods proposed in [9,10,13] have been implemented (2D fuzzy (spatial), directional fuzzy filter and isotropic fuzzy filter). Only the non-edge pixels (with G>180 in [9]) are filtered to avoid destroying the real edges of the image. All the parameters in Section 5 are chosen experimentally using a range of sequences in order to achieve the best visual quality.

For the directional fuzzy filter, parameter σ in (19) is chosen to effectively remove the overall artifacts [9]. Parameter γ controls the balance between removing the artifacts in the flat regions and keeping the details in the high activity regions. Parameters α and β are used to adjust the relative filtering strength between the gradient and tangent directions of the edges. These parameters are defined in [9], and we have determined their values experimentally, resulting in σ_A=15, α=0.5, and β=3.5 in (19) and γ=0.5 as defined in [9]. In the proposed method, parameter σ is chosen based on Fig. 4. The set ω of neighboring pixels and the spatial window W size are set to 5x5. Several *CIF* resolution video sequences have been compressed using motion *JPEG* with scaling factor 4 for the quantization step. The test set used includes different types of images from the following video sequences: Silent, Foreman, Mobile, Paris, News, and Mother (50 frames used from each sequence).

To demonstrate the visual quality, results obtained with different artifact removal techniques on a compressed and zoomed frame in Mobile are shown in Fig. 6. For this simulation, the spread parameter has been calculated using the method in Fig. 4. Compared to the compressed full frame in Fig. 6b (*PSNR*=21.253db, *MSSIM*=0.460) and the enhanced image using 2D fuzzy filter in Fig. 6c (*PSNR*=21.775, *MSSIM*=0.482), the enhanced image using the proposed method in Fig. 6d (*PSNR*=21.961, *MSSIM*=0.491) achieved a clear improvement in terms of *PSNR* and *MSSIM*. Comparing to the compressed image in Fig 6b, the 2D fuzzy method can remove most of the blocking and ringing artifacts. However, it in turn introduces other artifacts, such as blurring at the dominant edges and loss of details. The frame enhanced with the proposed algorithm achieves better quality, compared to the other fuzzy methods. It efficiently removes the blocking and ringing artifacts, but still keeps the details and the sharpness of the edges.

The average *PSNR* and *MSSIM* values for different sequences are listed in Tables 1 and 2, respectively. The average improvement in *PSNR* and *MSSIM* when using the proposed method, Isotropic [13], Directional [10], and 2D Fuzzy [9] are (0.821db, 0.076), (0.301db, 0.044), (0.608db, 0.056) and (0.633db, 0.062).

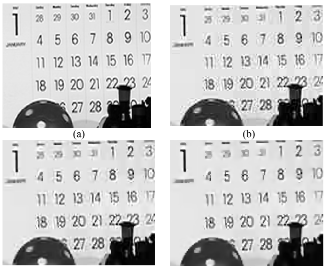
Table 1: Comparison of *PSNR* in dB for Different methods

PSNR	JPEG	2D Fuzzy	Directional	Isotropic	Proposed
News	27.671	28.091	27.855	27.961	28.265
Silent	27.321	27.812	27.985	27.536	28.164
Foreman	28.267	28.881	28.908	28.427	29.195
Mobile	21.253	21.755	21.752	21.283	21.961
Mother	30.811	31.824	31.652	31.512	31.871

Paris	23.432	24.191	24.252	23.843	24.226
Ave Gain		**0.633**	**0.608**	**0.301**	**0.821**

Table 2: Comparison of *MSSIM* for different methods

MSSIM	JPEG	2 D Fuzzy	Directiona l	Isotropic	Proposed
News	0.532	0.554	0.546	0.542	0.560
Silent	0.441	0.495	0.497	0.473	0.516
Foreman	0.512	0.556	0.543	0.521	0.578
Mobile	0.460	0.482	0.480	0.475	0.491
Mother	0.534	0.541	0.538	0.529	0.551
Paris	0.484	0.518	0.506	0.495	0.528
Ave Gain		**0.062**	**0.056**	**0.044**	**0.076**



(a) (b)

(c) (d)

Fig. 6. JPEG zoomed images for comparison: (a) Original Frame (b) Compressed; (c) 2D Fuzzy (d) Proposed Method

5.2 Enhancement of Compressed Video Sequences

To evaluate the proposed method for video sequences (Fig.5), different methods have been applied to decoded *MJPEG* and H.264 sequences. In the MJPEG codec, each frame is compressed separately using the JPEG standard compression. For practical reasons, the scaling function in (18) is set as a constant, $K[i+i',j+j']=1$, and $\sigma=\sigma_A$ is given by Fig. 4. The sizes of ω and the spatial window are 5x5 pixels.

Figures 7 and 8 compare the *PSNR* and *MSSIM* values of the methods tested for 50 frames of the Mobile sequence. The plots clearly demonstrate that the proposed Fuzzy-PDE filter achieves consistent *PSNR* gain of about 0.53dB on average, relative to the compressed frames, and about 0.12dB compared to the frames enhanced with the conventional fuzzy spatial filter method [10]. The respective *MSSIM* gains are about 0.02 and 0.05.

The visual improvement obtained with the proposed scheme is much more noticeable when the processed frames are played in a sequence, as the proposed method produces a video of smoother quality with significantly reduced artifacts.

In order to demonstrate that the proposed method is beneficial also for more efficient video compression methods, further experiments have been performed using H.264 video compression. The Foreman sequence was compressed with the prediction structure of IPPP at a bitrate of 132 Kbps. Two different types of experiments have been performed on the H.264 video sequences. In the first experiment, the in-loop de-blocking filter was disabled. In the second experiment, the algorithm was applied with in-loop filtering enabled. The adaptive spread parameter σ is applied to each block and the offset γ [9] was set to 0.5. These parameters have been chosen experimentally to get the best visual quality for a wide range of sequences.

In the first experiment, the proposed algorithm improves PSNR and MSSIM (averages are 34.912 db, 0.9834), compare to PSNR and MSSIM with fuzzy method (averages are 34.623 db, 0.9812) and disable in-loop filtering (averages are 34.248 db, 0.979).

In the second experiment, the proposed method improves *PSNR* and *MSSIM* (averages are 34.881 db, 0.9864), compared to the *PSNR* and *MSSIM* with the fuzzy method (averages are 34.766 db, 0.9814) and in-loop filtering (averages are 34.38 db, 0.979). This improvement is consistent for the Foreman sequence, which is verified by the *PSNR* and *MSSIM* curves shown in Figs. 9 and 10. Visual analysis shows that compared to in-loop filtering, the fuzzy method and the proposed method both significantly reduce the ringing artifacts close to edges. However, the proposed method maintains the sharpness of edges better than the fuzzy method. This improvement indicates a more pleasant visual appearance.

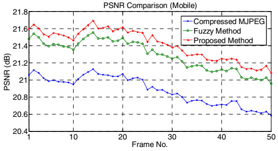


Fig. 7. Comparison on PSNR of simulated methods.

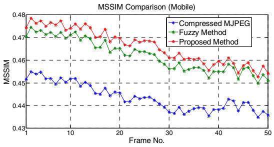


Fig. 8. Comparison on MSSIM of simulated methods.

The enhanced video sequence using the proposed method has fewer artifacts than the compressed sequence enhanced with the fuzzy method and the in-loop de-blocking filter. The *PSNR* improvement with the proposed algorithm applied to the Foreman sequence compressed with different bitrates is shown in Fig. 11. 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 analysis indicates higher quality with the proposed method.

Another benefit of our method is that it does not require a motion compensation stage and spatiotemporal filtering. This is why its computational complexity is reasonably low.

6. CONCLUSIONS

In this paper, we have proposed an effective algorithm for image and video artifact removal using an adaptive fuzzy filter and directional anisotropic diffusion. This novel method overcomes the limitations of the conventional nonlinear filters by taking pixel's activity and the direction between pixels both into account. It has been shown that the proposed algorithm improves the visual quality of compressed images and videos in terms of *PSNR* and *MSSIM*, compared to existing approaches. The proposed adaptive scheme can be applied to different image and video compression standards, such as *JPEG, MJPEG* and *H.264*.

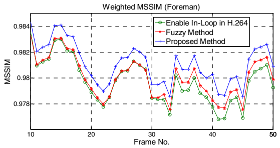


Fig. 9. Comparison on Weighed MSSIM of simulated methods.

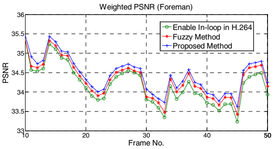


Fig. 10. Comparison on Weighed PSNR of simulated methods.

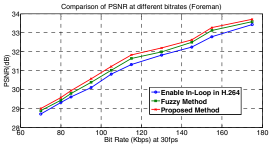


Fig. 11. Comparison of PSNR with different bit-rates.

8. REFERENCES

[1] T. Chen, H. Wu, and B. Qiu, "Adaptive Postfiltering of Transform Coefficients for the Reduction of Blocking Artifacts," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, pp. 594–602, May 2001.

[2] S. Liu and A. Bovik, "Efficient DCT-domain Blind Measurement and Reduction of Blocking Artifacts," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, pp. 1139–1149, Dec. 2002.

[3] Y. Yang and N. P. Galatsanos, "Removal of Compression Artifacts using Projections Onto Convex Sets and Line Process Modeling", *IEEE Transactions on Image Processing* , Vol. 6, pp. 1345-1357,1997.

[4] S. Oguz, Y. Hu, and T. Nguyen, "Image Coding Ringing Artifact Reduction using Morphological Post-filtering," *Proc. IEEE Int. Work. Multimedia Signal Proc.*, pp. 628–633, 1998.

[5] H. Kong, Y. Nie, A. Vetro, H. Sun, and K. Barner, "Adaptive Fuzzy Post-filtering for Highly Compressed Video," *Proc. of IEEE Int. Conf. Image Proc.*, pp. 1802–1806, 2004.

[6] S. Westen, R. Lagendijk, and J. Biemond, "Adaptive Spatial Noise Shaping for DCT based Image Compression," *IEEE Int. Conf. Acoustics, Speech and Signal Processing*, vol. 4, pp.2124–2127. May 1996,

[7] X. Fan, W. Gao, Y. Lu, and D. Zhao, "Flicking Reduction in All Intra Frame Coding," *Joint Video Team of ISO/IEC MPEG and ITU-T VCEG, JVT-E070*, 2002.

[8] A. Leontaris, Y. Tonomura, T. Nakachi, and P. Cosman, "Flicker Suppression in JPEG2000 using Segmentation-based Adjustment of Block Truncation Lengths," *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Proc. (ICASSP)*, vol. 1, pp. 1117–1120, 2007.

[9] D.T.Vo, T.Q.Nguyen, S.Yea, A.Vetro , "Adaptive Fuzzy Filtering for Artifact Reduction in Compressed Images and Videos", *IEEE Transactions on Image Processing*, Vol. 18, pp. 1057-7149, 2009.

[10] D. Vo and T. Nguyen, "Directional Motion-compensated Spatio-temporal Fuzzy Filtering for Quality Enhancement of Compressed Video Sequences," *IEEE Int. Conf. on Image Proc.* (ICIP), 2008.

[11] P. Perona , and J. Malik, "Scale-space and Edge Detection using Anisotropic Diffusion," *IEEE Trans. On Pattern Analysis and Machine Intel.*, Vol. 127, pp. 629-639,1990.

[12] F. Catte, P. L. Lions, J. M. Morel and T. Coll, "Image Selective Smoothing and Edge Detection by Nonlinear Diffusion", *SIAM J. Num. Anal.,* vol. 29, no. 1, pp. 182-193, 1992.

[13] H. Kong, Y. Nie, A. Vetro, H. Sun, and K. Barner, "Adaptive Fuzzy Post-Filtering for Highly Compressed Video," in *Proc. IEEE Int. Conf. Image Proc* (ICIP), pp. 1802–1806, 2004.

[14] Y. Nie and K. Barner, "The Fuzzy Transformation and its Application in Image Processing," *IEEE Trans. Image Process.*, vol. 15, no. 4, pp.910–927, 2006.

[15] F.D. Simone, D. Ticca, F. Dufaux, M. Ansorge, T. Ebrahimi, and A.G. Tescher, "A Comparative Study of Color Image Compression Standards using Perceptually Driven Quality Metrics", *SPIE Optics and Photonics, Applications of Digital Image Proc. XXXI*, vol. 70730Z, 2008.

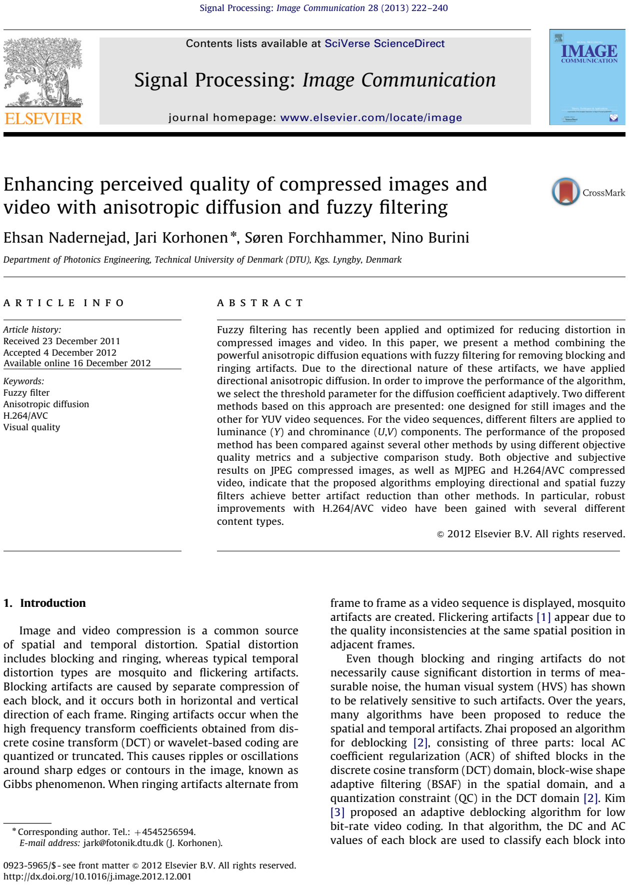
[16] E. Nadernejad, H. Hassanpour, "A Comparison and Analysis of Different PDE-based Approaches for Image Enhancement". *Int. Conf. on Signal Proc. and Comm. Syst.* (ICSPCS), Australia, 2007.

[17] <http://www.imageprocessingplace.com/root_files_V3/image_databases.htm>.

A.2 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: [2]



Signal Processing: *Image Communication* 28 (2013) 222–240

Contents lists available at SciVerse ScienceDirect

Signal Processing: *Image Communication*

journal homepage: www.elsevier.com/locate/image

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
Available online 16 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.

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: jark@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

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 postprocessing. 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

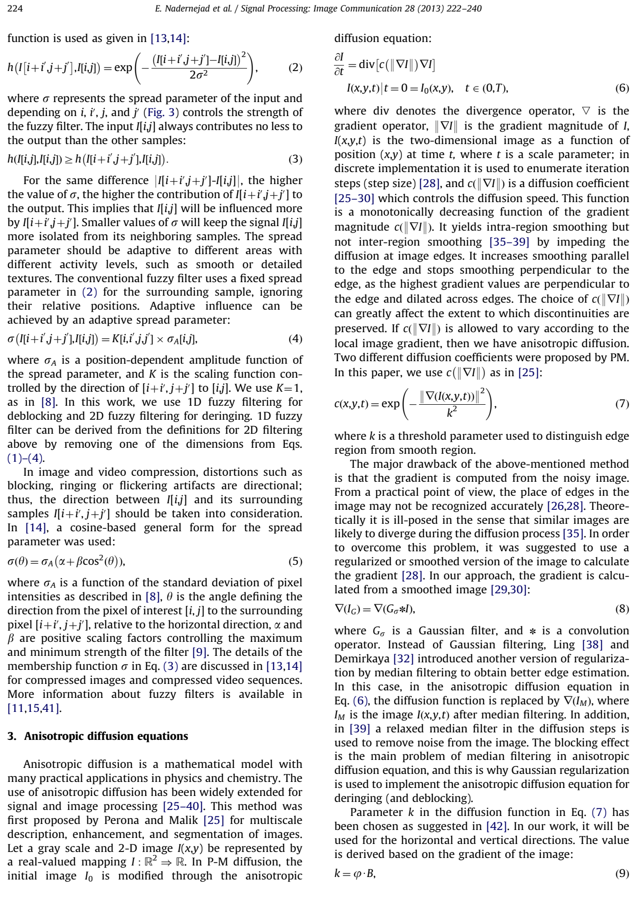
Fuzzy filtering has been used with success in deblocking and deringing of compressed images and videos [11,13,14,41]. Fuzzy filters, such as those described in [13,14], are designed for similar purposes as median filters [15] or rank condition rank selection filters [41]. In this section, we describe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

$$\hat{I}[i,j] = \frac{\sum_{[i',j'] \in \omega} h\left(I[i+i',j+j'],I[i,j]\right) I[i+i',j+j']}{\sum_{[i',j'] \in \omega} h\left(I[i+i',j+j'],I[i,j]\right)}, \quad (1)$$

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

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



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

$$h\left(I\left[i+i',j+j'\right],I[i,j]\right)=\exp\left(-\frac{\left(I[i+i',j+j']-I[i,j]\right)^2}{2\sigma^2}\right), \tag{2}$$

where σ represents the spread parameter of the input and depending on i, i', j, and j' (Fig. 3) controls the strength of the fuzzy filter. The input $I[i,j]$ always contributes no less to the output than the other samples:

$$h(I[i,j],I[i,j])\geq h\left(I[i+i',j+j'],I[i,j]\right). \tag{3}$$

For the same difference $\left|I[i+i',j+j']\text{-}I[i,j]\right|$, the higher the value of σ, the higher the contribution of $I[i+i',j+j']$ to the output. This implies that $I[i,j]$ will be influenced more by $I[i+i',j+j']$. Smaller values of σ will keep the signal $I[i,j]$ 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:

$$\sigma\left(I[i+i',j+j'],I[i,j]\right)=K[i,i',j,j']\times\sigma_A[i,j], \tag{4}$$

where σ_A is a position-dependent amplitude function of the spread parameter, and K is the scaling function controlled by the direction of $[i+i',j+j']$ to $[i,j]$. 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 $I[i,j]$ and its surrounding samples $I[i+i',j+j']$ should be taken into consideration. In [14], a cosine-based general form for the spread parameter was used:

$$\sigma(\theta)=\sigma_A\left(\alpha+\beta\cos^2(\theta)\right), \tag{5}$$

where σ_A is a function of the standard deviation of pixel intensities as described in [8], θ is the angle defining the direction from the pixel of interest $[i,j]$ to the surrounding pixel $[i+i',j+j']$, relative to the horizontal direction, α and β are positive scaling factors controlling the maximum and minimum strength of the filter [9]. The details of the membership function σ 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:

$$\frac{\partial I}{\partial t}=\operatorname{div}\left[c\left(\|\nabla I\|\right)\nabla I\right]$$
$$I(x,y,t)\big|t=0=I_0(x,y),\quad t\in(0,T), \tag{6}$$

where div denotes the divergence operator, $\bigtriangledown$ 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 $c\left(\|\nabla I\|\right)$ as in [25]:

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

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

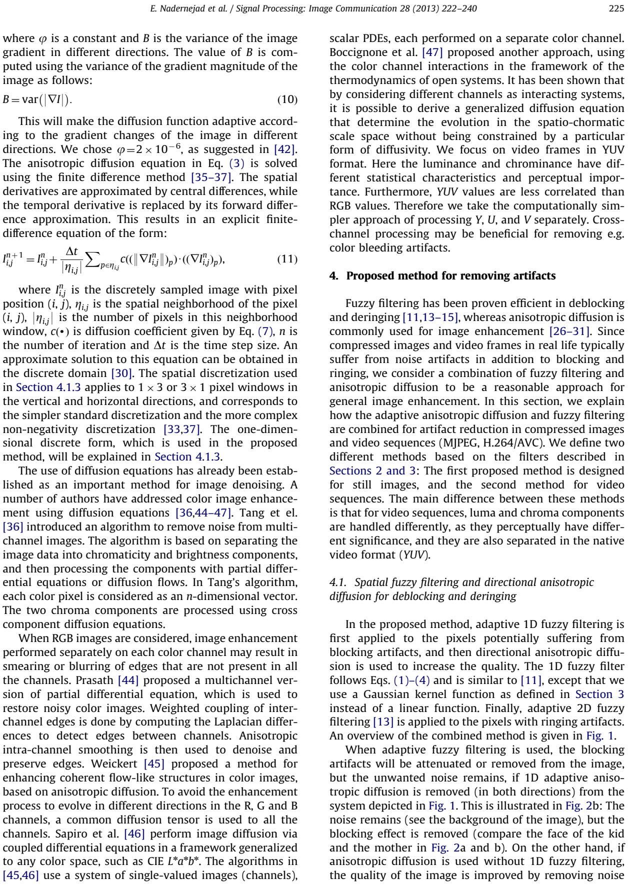
The major drawback of the above-mentioned method is that the gradient is computed from the noisy image. From a practical point of view, the place of edges in the image may not be recognized accurately [26,28]. Theoretically it is ill-posed in the sense that similar images are likely to diverge during 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]:

$$\nabla(I_G)=\nabla(G_\sigma * I), \tag{8}$$

where G_σ 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:

$$k=\varphi\cdot B, \tag{9}$$



where φ 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:

$$B=\mathrm{var}(|\nabla I|). \tag{10}$$

This will make the diffusion function adaptive according to the gradient changes of the image in different directions. We chose $\varphi=2\times10^{-6}$, 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:

$$I_{i,j}^{n+1}=I_{i,j}^{n}+\frac{\Delta t}{|\eta_{i,j}|}\sum\nolimits_{p\in\eta_{i,j}}c((\|\nabla I_{i,j}^{n}\|)_p)\cdot((\nabla I_{i,j}^{n})_p), \tag{11}$$

where $I_{i,j}^{n}$ is the discretely sampled image with pixel position (i, j), $\eta_{i,j}$ is the spatial neighborhood of the pixel (i, j), $|\eta_{i,j}|$ is the number of pixels in this neighborhood window, $c(\bullet)$ is diffusion coefficient given by Eq. (7), n is the number of iteration and Δt 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 × 3 or 3 × 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 el. [36] introduced an algorithm to remove noise from multichannel 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 interchannel 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-chormatic 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 (MJPEG, 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

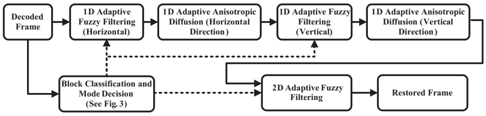

Decoded Frame
1D Adaptive Fuzzy Filtering (Horizontal)
1D Adaptive Anisotropic Diffusion (Horizontal Direction)
1D Adaptive Fuzzy Filtering (Vertical)
1D Adaptive Anisotropic Diffusion (Vertical Direction)
Block Classification and Mode Decision (See Fig. 3)
2D Adaptive Fuzzy Filtering
Restored Frame

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

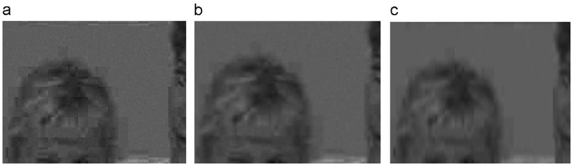


Fig. 2. The effect of removing the adaptive fuzzy filter and adaptive anisotropic diffusion: (a) the original compressed image, (b) 1D adaptive fuzzy filtering (both directions) and (c) 1D adaptive anisotropic diffusion (both directions).

(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 × 3 pixel blocks, denoted as *STD*(*I*(*i*,*j*)) when the center pixel of the block is at position [*i*,*j*]. The classification is based on comparing the *STD* value in a 3 × 3 windows around each pixel and the value of the maximum *STD* (MaxSTD) in a 4 × 4 block (with H.264/AVC) or an 8 × 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.

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

Fig. 3 shows the flowchart of the proposed algorithm to find the spread parameter (σ) 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 × 4 (for H.264) or 8 × 8 pixel blocks, and then classified in five categories. In contrast, four categories for 8 × 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 × 8 or 4 × 4 block, which are the relevant sizes for most of the practical block-based codecs employing transform of 8 × 8 (as JPEG and MJPEG) or 4 × 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_7 and X_0 (8 × 8 pixels) or Y_3 and X_0 (4 × 4 pixels). The absolute differences between pixels are denoted using R and L, right and left of the border, respectively (Fig. 3). Then, if MAX(L_1, L_2, L_3, L_4) < G_0 or MAX(R_1, R_2, R_3, R_4) < G_0 (in case of JPEG) or MAX (L_1, L_2, L_3) < G_0 or 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 σ 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

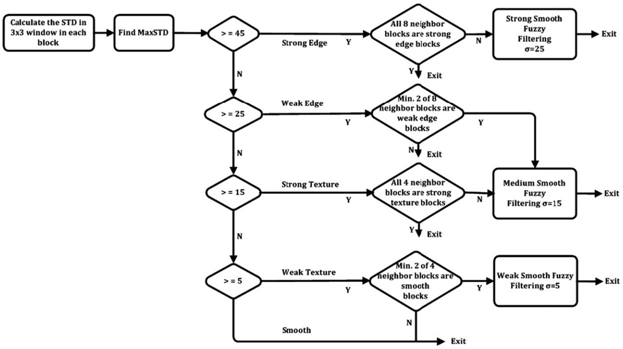

Calculate the STD in 3x3 window in each block
Find MaxSTD
> = 45
Strong Edge
All 8 neighbor blocks are strong edge blocks
Strong Smooth Fuzzy Filtering σ=25
> = 25
Weak Edge
Min. 2 of 8 neighbor blocks are weak edge blocks
> = 15
Strong Texture
All 4 neighbor blocks are strong texture blocks
Medium Smooth Fuzzy Filtering σ=15
> = 5
Weak Texture
Min. 2 of 4 neighbor blocks are smooth blocks
Weak Smooth Fuzzy Filtering σ=5
Smooth
Y
N
Exit

Fig. 3. Adaptive spread parameter (σ) selection for a block.

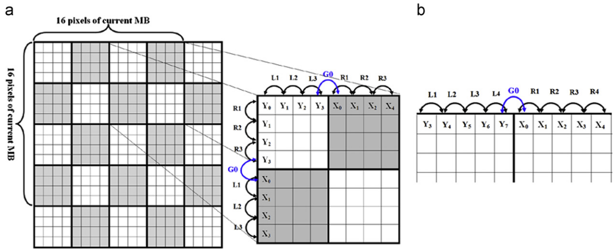
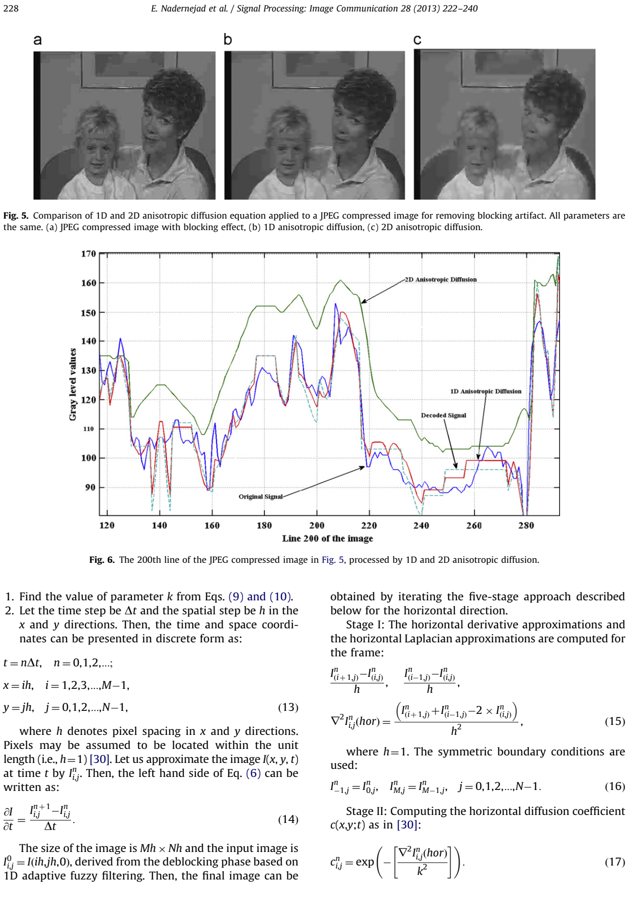


Fig. 4. Detecting the vertical boundary gap in a row across the vertical block boundary (a) for H.264/AVC video sequences and (b) for JPEG images and MJPEG video sequences.

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.



a
b
c

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.

Gray level values
170
160
150
140
130
120
110
100
90
120
140
160
180
200
220
240
260
280
Line 200 of the image
2D Anisotropic Diffusion
1D Anisotropic Diffusion
Decoded Signal
Original Signal

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 Δt 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:

$$t=n\Delta t,\quad n=0,1,2,\ldots;$$

$$x=ih,\quad i=1,2,3,\ldots,M-1,$$

$$y=jh,\quad j=0,1,2,\ldots,N-1, \tag{13}$$

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 $I(x, y, t)$ at time t by $I^n_{i,j}$. Then, the left hand side of Eq. (6) can be written as:

$$\frac{\partial I}{\partial t}=\frac{I^{n+1}_{i,j}-I^n_{i,j}}{\Delta t}. \tag{14}$$

The size of the image is $Mh\times Nh$ and the input image is $I^0_{i,j}=I(ih,jh,0)$, 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:

$$\frac{I^n_{(i+1,j)}-I^n_{(i,j)}}{h},\quad \frac{I^n_{(i-1,j)}-I^n_{(i,j)}}{h},$$

$$\nabla^2 I^n_{i,j}(hor)=\frac{\left(I^n_{(i+1,j)}+I^n_{(i-1,j)}-2\times I^n_{(i,j)}\right)}{h^2}, \tag{15}$$

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

$$I^n_{-1,j}=I^n_{0,j},\quad I^n_{M,j}=I^n_{M-1,j},\quad j=0,1,2,\ldots,N-1. \tag{16}$$

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

$$c^n_{i,j}=\exp\left(-\left[\frac{\nabla^2 I^n_{i,j}(hor)}{k^2}\right]\right). \tag{17}$$

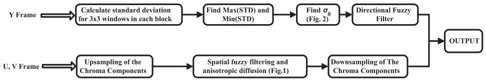

Y Frame
Calculate standard deviation for 3x3 windows in each block
Find Max(STD) and Min(STD)
Find σ_0 (Fig. 2)
Directional Fuzzy Filter
OUTPUT
U, V Frame
Upsampling of the Chroma Components
Spatial fuzzy filtering and anisotropic diffusion (Fig.1)
Downsampling of The Chroma Components

Fig. 7. Proposed method for video frames.

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

$$d_{i,j}^{n}=\frac{1}{h^{2}}\left[c_{i+1,j}^{n}\left(I_{i+1,j}^{n}-I_{i,j}^{n}\right)+c_{i-1,j}^{n}\left(I_{i-1,j}^{n}-I_{i,j}^{n}\right)\right], \tag{18}$$

with the symmetric boundary conditions:

$$d_{-1,j}^{n}=d_{0,j}^{n},\quad d_{M,j}^{n}=d_{M-1,j}^{n},\quad j=0,1,2,\ldots,N-1. \tag{19}$$

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

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

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

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 tradeoff 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 3×3 window was computed around each pixel and then the MaxSTD in each 4×4 block or 8×8 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 σ. 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 (σ=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 4×4 blocks and 8×8 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].

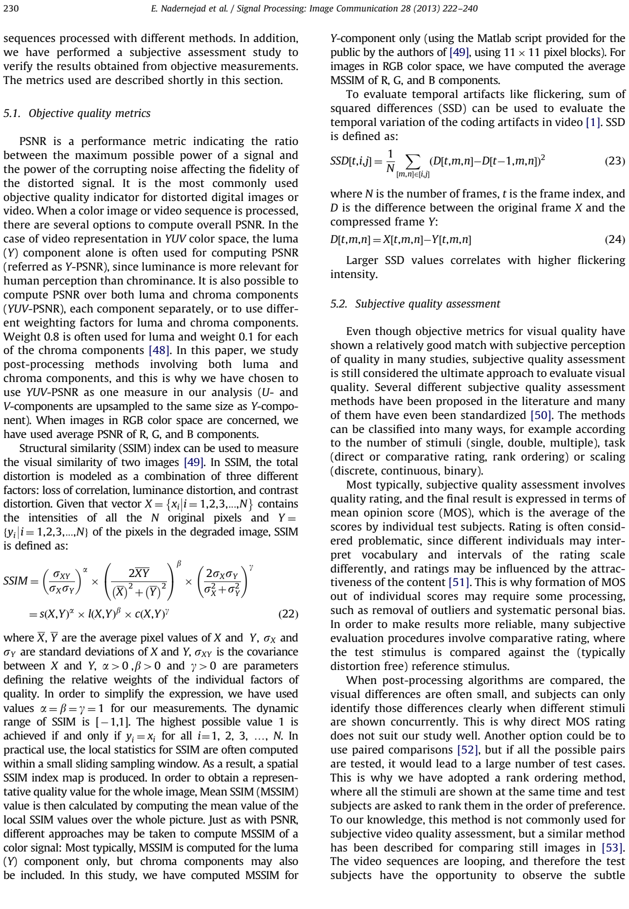
In the proposed algorithm, we have examined different designs to find the best algorithm for deblocking and deringing in luminance and chrominance channels, and a hybrid algorithm where anisotropic diffusion is combined with fuzzy filtering is proposed, as shown in Fig. 7. In this algorithm, the chrominance 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 σ_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 σ_A obtained from:

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

where STD_{min} and STD_{max} are the minimum and maximum standard deviations found in the current frame, and γ 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



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 $X=\{x_i|i=1,2,3,\ldots,N\}$ contains the intensities of all the N original pixels and $Y=\{y_i|i=1,2,3,\ldots,N\}$ of the pixels in the degraded image, SSIM is defined as:

$$\begin{aligned} SSIM &= \left(\frac{\sigma_{XY}}{\sigma_X\sigma_Y}\right)^{\alpha} \times \left(\frac{2\overline{XY}}{(\overline{X})^2+(\overline{Y})^2}\right)^{\beta} \times \left(\frac{2\sigma_X\sigma_Y}{\sigma_X^2+\sigma_Y^2}\right)^{\gamma} \\ &= s(X,Y)^{\alpha} \times l(X,Y)^{\beta} \times c(X,Y)^{\gamma} \end{aligned} \tag{22}$$

where $\overline{X}$, $\overline{Y}$ are the average pixel values of X and Y, σ_X and σ_Y are standard deviations of X and Y, σ_{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 $i=1, 2, 3, \ldots, N$. 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 × 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 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 el. [2], Kim [3], Chen et al. [4], Liu et el. [5], Vo et el. (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:

$$\mathrm{PSNR}(n+1) < \mathrm{PSNR}(n), \tag{25}$$

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 γ defined in Eq. (21) controls the balance between artifact removal in the flat regions and keeping the details in the high activity regions. Parameters α and β 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 σ_0 computed as shown in the flowchart in Fig. 3. The set ω of neighboring pixels in Eq. (1) and the spatial window W size are set to 5×5. 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	**26.19** (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	**26.43** (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	**26.26** (9)
Mobile	21.25	21.17	21.33	21.48	21.51	21.73	21.76	21.65	21.75	21.62	21.28	**21.96**	**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	**28.90** (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	**22.71** (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	**0.847**

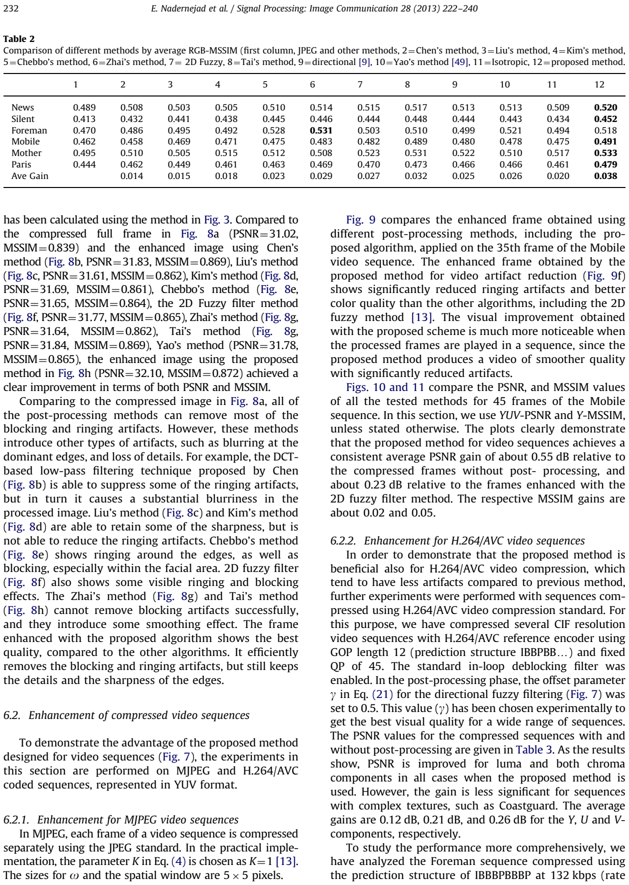


Table 2
Comparison of different methods by average RGB-MSSIM (first column, JPEG and other methods, 2=Chen's method, 3=Liu's method, 4=Kim's method, 5=Chebbo's method, 6=Zhai's method, 7= 2D Fuzzy, 8=Tai's method, 9=directional [9], 10=Yao's method [49], 11=Isotropic, 12=proposed method.

	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	**0.520**
Silent	0.413	0.432	0.441	0.438	0.445	0.446	0.444	0.448	0.444	0.443	0.434	**0.452**
Foreman	0.470	0.486	0.495	0.492	0.528	**0.531**	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	**0.491**
Mother	0.495	0.510	0.505	0.515	0.512	0.508	0.523	0.531	0.522	0.510	0.517	**0.533**
Paris	0.444	0.462	0.449	0.461	0.463	0.469	0.470	0.473	0.466	0.466	0.461	**0.479**
Ave Gain		0.014	0.015	0.018	0.023	0.029	0.027	0.032	0.025	0.026	0.020	**0.038**

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. 8g, PSNR=31.84, MSSIM=0.869), Yao's method (PSNR=31.78, MSSIM=0.865), the enhanced image using the proposed method in Fig. 8h (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 ω and the spatial window are 5×5 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 γ in Eq. (21) for the directional fuzzy filtering (Fig. 7) was set to 0.5. This value (γ) 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 IBBBPBBBP at 132 kbps (rate

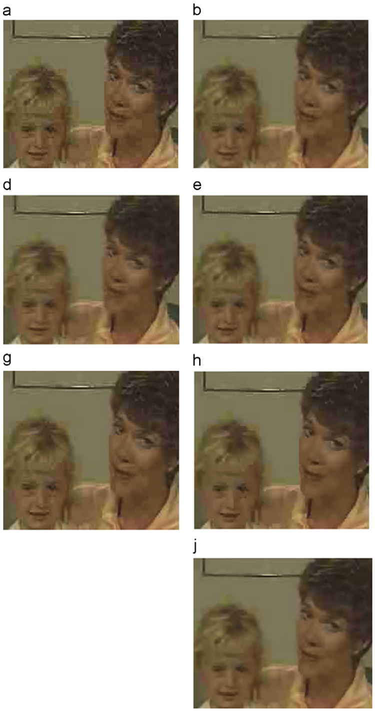


Fig. 8. The comparison of filtered results on zoomed images: (a) the compressed image-4Q (PSNR=31.02, MSSIM=0.839), (b) Chen's method (PSNR=31.83, MSSIM=0.869); (c) Liu's method (PSNR=31.61, MSSIM=0.862), (d) Kim's method (PSNR=31.69, MSSIM=0.861), (e) Chebbo's method (PSNR=31.65, MSSIM=0.864), (f) 2D 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

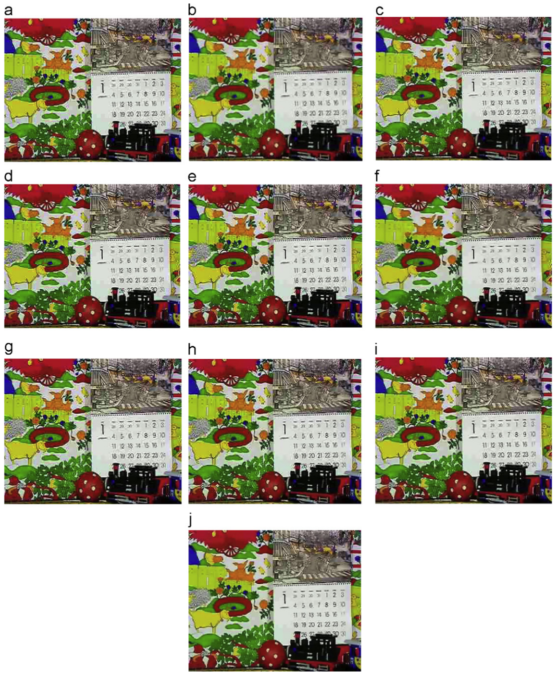
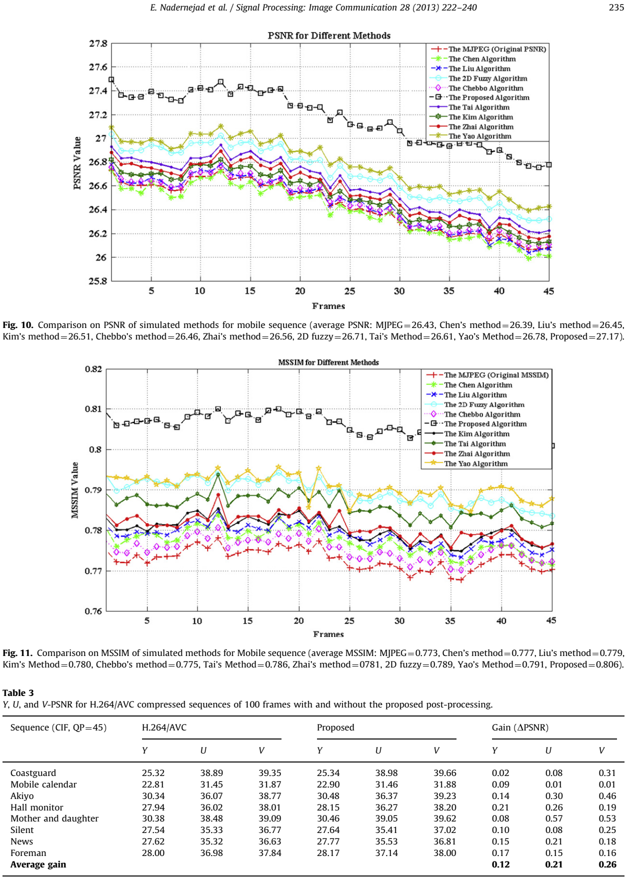


Fig. 9. Comparison of filter results for MJPEG sequences (35 th frame). (a) Compressed (PSNR=23.03, MSSIM =0.786), (b) Chen's method (PSNR=22.58, MSSIM=0.774); (c) Liu's method (PSNR=22.88, MSSIM=0.782), (d) Kim's method (PSNR=23.28, MSSIM=0.794), (e) Chebbo's method (PSNR=23.13, MSSIM=0.793), (f) 2D fuzzy filter method (PSNR=23.26, 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



E. Nadernejad et al. / Signal Processing: Image Communication 28 (2013) 222–240 235

Fig. 10. Comparison on PSNR of simulated methods for mobile sequence (average PSNR: MJPEG=26.43, Chen's method=26.39, Liu's method=26.45, Kim's method=26.51, Chebbo'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, Chebbo's method=0.775, Tai's Method=0.786, Zhai's method=0781, 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	39.66	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
Average gain							**0.12**	**0.21**	**0.26**

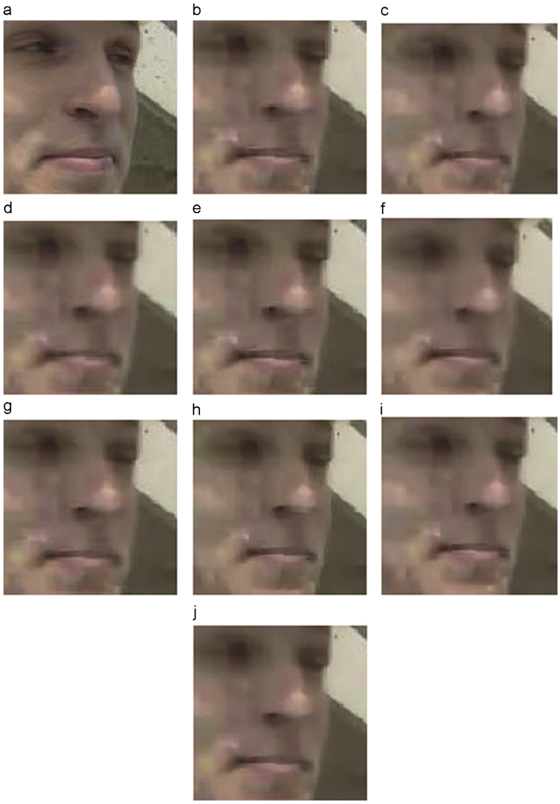


Fig. 12. Comparison of filter results for H.264/AVC sequences (frame no. 3 zoomed): (a) original frame, (b) compressed with in-loop filtering (PSNR=32.30, MSSIM=0.869), (c) Kim's method (PSNR=32.32, MSSIM=0.872), (d) Chebbo's method (PSNR=32.32, MSSIM=0.870), (e) 5/3 Haar MCTF method (PSNR=32.11, MSSIM=0.873), (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=0.876).

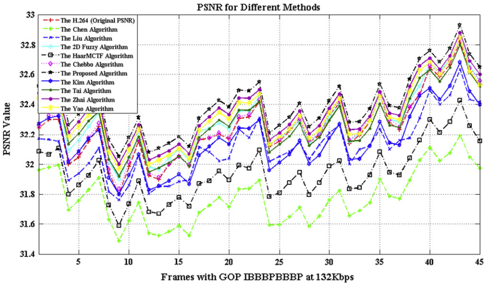


Fig. 13. Comparison on PSNR of simulated methods for Foreman sequence.

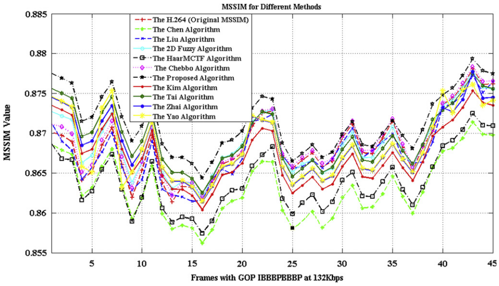


Fig. 14. Comparison on MSSIM of simulated methods for Foreman sequence.

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.

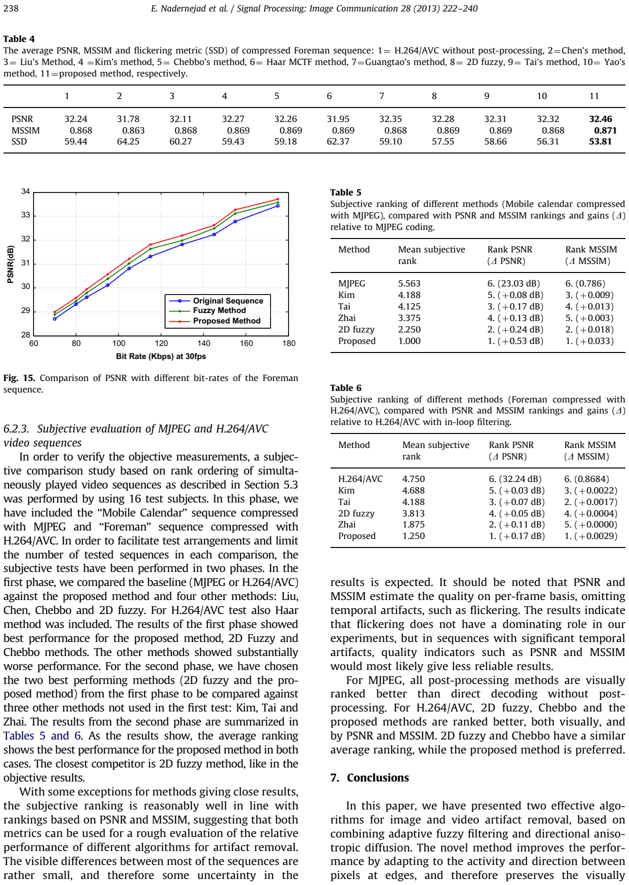


Table 4
The average PSNR, MSSIM and flickering metric (SSD) of compressed Foreman sequence: 1= H.264/AVC without post-processing, 2=Chen's method, 3= Liu's Method, 4 =Kim's method, 5= Chebbo's method, 6= Haar MCTF method, 7=Guangtao's method, 8= 2D fuzzy, 9= Tai's method, 10= Yao's method, 11=proposed 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	**32.46**
MSSIM	0.868	0.863	0.868	0.869	0.869	0.869	0.868	0.869	0.869	0.868	**0.871**
SSD	59.44	64.25	60.27	59.43	59.18	62.37	59.10	57.55	58.66	56.31	**53.81**

Fig. 15. Comparison of PSNR with different bit-rates of the Foreman sequence.

Table 5
Subjective ranking of different methods (Mobile calendar compressed with MJPEG), compared with PSNR and MSSIM rankings and gains (Δ) relative to MJPEG coding.

Method	Mean subjective rank	Rank PSNR (Δ PSNR)	Rank MSSIM (Δ MSSIM)
MJPEG	5.563	6. (23.03 dB)	6. (0.786)
Kim	4.188	5. (+0.08 dB)	3. (+0.009)
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 (Δ) relative to H.264/AVC with in-loop filtering.

Method	Mean subjective rank	Rank PSNR (Δ PSNR)	Rank MSSIM (Δ 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)

6.2.3. Subjective evaluation of MJPEG and H.264/AVC video sequences

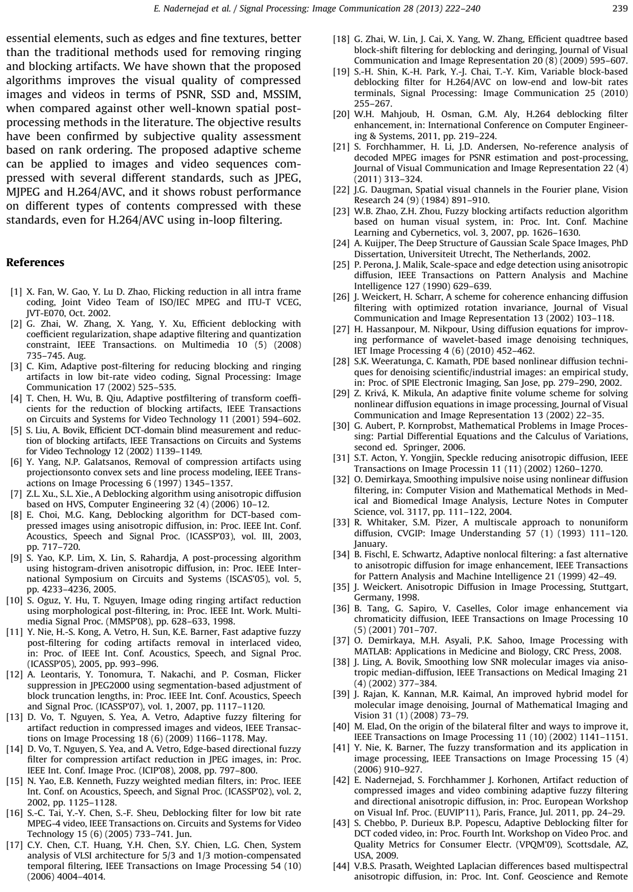
In order to verify the objective measurements, a subjective comparison study based on rank ordering of simultaneously played video sequences as described in Section 5.3 was performed by using 16 test subjects. In this phase, we have included the “Mobile Calendar” sequence compressed with MJPEG and “Foreman” 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 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



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, Flicking 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) 594–602.

[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 projectionsonto 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 oding ringing artifact reduction using morphological post-filtering, in: Proc. IEEE Int. Work. Multimedia Signal Proc. (MMSP'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. Leontaris, Y. Tonomura, 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. Kenneth, 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 127 (1990) 629–639.

[26] J. Weickert, H. Scharr, 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. Weeratunga, 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. Krivá, 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 Processin 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 for 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.

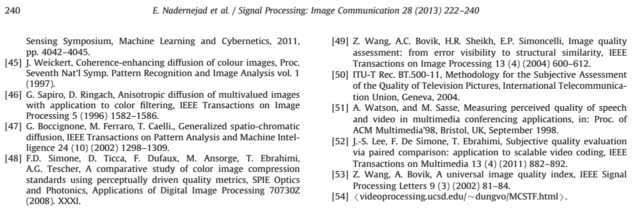
[40] M. Elad, On the origin of the bilateral filter and ways to improve it, IEEE Transactions on Image Processing 11 (10) (2002) 1141–1151.

[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. Durieux B.P. Popescu, Adaptive Deblocking filter for DCT coded video, in: Proc. Fourth Int. Workshop on Video Proc. and Quality Metrics for Consumer Electr. (VPQM'09), Scottsdale, AZ, USA, 2009.

[44] V.B.S. Prasath, Weighted Laplacian differences based multispectral anisotropic diffusion, in: Proc. Int. Conf. Geoscience and Remote



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⟩.

A.3 Adaptive De-Blocking and De-Ringing 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)*, May 2013, pp. 2508–2512

Reference: [3]

ADAPTIVE DEBLOCKING AND DERINGING OF H.264/AVC VIDEO SEQUENCES

Ehsan Nadernejad, Nino Burini, Søren Forchhammer

Department of Photonics Engineering, Technical University of Denmark, Ørsteds 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, blockwise 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.

978-1-4799-0356-6/13/$31.00 ©2013 IEEE

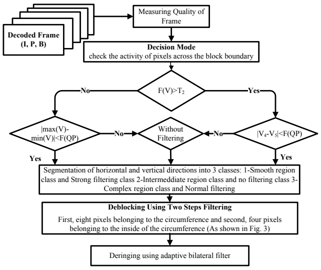

Decoded Frame
(I, P, B)
Measuring Quality of Frame
Decision Mode
check the activity of pixels across the block boundary
F(V)>T₂
No
Yes
|max(V)-min(V)|<F(QP)
No
Without Filtering
No
|V₄-V₅|<F(QP)
Yes
Yes
Segmentation of horizontal and vertical directions into 3 classes: 1-Smooth region class and Strong filtering class 2-Intermeddiate region class and no filtering class 3-Complex region class and Normal filtering
Deblocking Using Two Steps Filtering
First, eight pixels belonging to the circumference and second, four pixels belonging to the inside of the circumference (As shown in Fig. 3)
Deringing using adaptive bilateral filter

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]:

$$Q_M = \sqrt{12 \times MSE} \tag{1}$$

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:

$$F(QP) = \sqrt{Q_M} \tag{2}$$

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 on the variation of activity in vertical and horizontal pixel vectors at each 4×4 block boundary, as shown in Fig. 3.

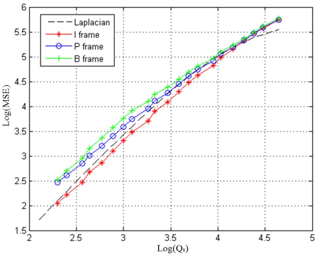


Fig. 2: MSE vs Q_s measured on mobcal (CIF); rate control is disabled, different QP values chosen for the different points [11].

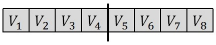


Fig. 3: Position of filtered pixels and pixel vector used for the decision mode process in horizontal direction. For vertical direction the pixel vector is the same.

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 4×4 block boundaries, as described in Fig. 3. The activity is as follows:

$$R(V) = \sum_{i=1}^{7} \phi(v_i - v_{i+1}), \tag{3}$$

where $0 \leq R(V) \leq 7$ and

$$\phi(\Delta) = \begin{cases} 1, & \text{if } |\Delta| < T_1 \\ 0, & \text{otherwise} \end{cases}, \tag{4}$$

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_4 and v_5) are considered. If the absolute

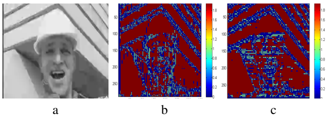


Fig. 4: Decision mode of the 25^{th} frame of the Foreman sequences, a) coded frame, b) horizontal direction modes, c) vertical direction modes.

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 mode 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 6×6 filtering window is centered at the intersection of four 4×4 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 4×4 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_{1u} in different modes ($\gg$ is the bitshift operator).

$$x_{1u} = \begin{cases} y_1 + 5x_1 + 3x_2 - y_2 \gg 3 & \text{if NW,} \\ 2a_7 + y_1 + 2x_1 + x_2 + y_2 + a_8 \gg 3 & \text{if NH,} \\ 4(x_1 + a_3) + 2(x_2 + y_2 + y_3) + y_1 + y_2 \gg 4 & \text{if WH,} \\ 4(a_2 + a_3 + a_4 + x_1) + 2(y_1 + x_5 + y_4 & \\ \quad +y_6) + a_7 + x_2 + y_2 + a_8 + y_3 + y_5 + & \text{if HH.} \\ \qquad\qquad x_3 + x_6 \gg 5 & \end{cases} \tag{5}$$

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_1	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_3	y_4	x_7	a_{10}
a_{11}	x_6	y_5	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 (6×6).

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:

$$y_{3u} = \begin{cases} 3x_5 + y_3 + y_7 - y_4 \gg 2 & \text{if NH} \\ 2x_5 + 5y_3 + 3y_4 - 2x_8 \gg 3 & \text{if NW} \\ 6x_5 + 4y_3 + 2y_4 + 4x_7 + 2y_5 - x_1 - x_3 \gg 4 & \text{if HW} \\ x_5 - x_7 + x_1 - x_3 + 4y_3 + 2y_5 + 2y_4 \gg 3 & \text{if WW} \\ 2(x_5 + x_7 + x_1 + x_3) + y_5 + 6y_3 + y_4 \gg 4 & \text{if HH} \end{cases} \tag{6}$$

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:

$$J(x) = \frac{1}{Z} \sum_{y \in \psi(x)} e^{\frac{-\|(y-x)\|^2}{2\sigma_d^2}} e^{\frac{-|(I(y)-I(x))|^2}{2\sigma_r^2}}, \tag{7}$$

where σ_d and σ_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:

$$Z = \sum_{y \in \psi(x)} e^{\frac{-\|(y-x)\|^2}{2\sigma_d^2}} e^{\frac{-|(I(y)-I(x))|^2}{2\sigma_r^2}}. \tag{8}$$

The behavior of the bilateral filter is determined by σ_d and σ_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 σ_d and σ_r values. In the proposed method, each 4×4 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 σ_d can be large, otherwise it should be small. Classification is done by computing the standard deviation (STD) in a 4×4 window around each pixel and comparing the maximum STD in each 4×4 block with a set of predetermined thresholds as follows:

$$\sigma_d = \begin{cases} StrongEdge, \sigma_d = 0.8 & \text{if MaxSTD} \in [35, \infty) \\ WeakEdge, \sigma_d = 1.8 & \text{if MaxSTD} \in [25, 35) \\ Texture, \sigma_d = 2.8 & \text{if MaxSTD} \in [15, 25) \\ Smooth, \sigma_d = 3.8 & \text{if MaxSTD} \in [0, 15) \end{cases} \quad (9)$$

The optimal σ_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 α is set to $1/3$ in each 4×4 block. The calculation of σ_d and σ_r are repeated for all blocks to obtain the block spatial map M_{σ_d} and the block intensity map M_{σ_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][1]. 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, Coastgard, 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 more effectively attenuated in both images, resulting in a better perceptual quality for the reconstructed video.

Table 1: The average results of post-processing H.264/AVC video test sequences using different algorithms.

Metric	PSNR		MSSIM	
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	
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

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.

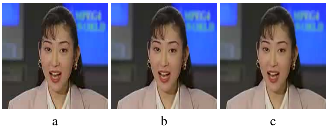


Fig. 6: The comparison of filter result on Akiyo (75^{th} frame), a) Compressed frame, b) In-loop filter, c) Proposed algorithm.

[1] The software for [9] was provided by Zhai

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 Bahadir 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 Bahadir 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 Ansorge, 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.4 Adaptive Local Backlight Dimming Algorithm Based on Local Histogram and Image Characteristics

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

Reference: [4]

Adaptive local backlight dimming algorithm based on local histogram and image characteristics

Ehsan Nadernejad[1], Nino Burini, Jari Korhonen, Søren Forchhammer, and Claire Mantel

Dept. of Photonics Eng., Tech. Univ. of Denmark, Ørsteds Plads, 2800 Kgs. Lyngby, Denmark;

ABSTRACT

Liquid Crystal Display (LCDs) with Light Emitting Diode (LED) backlight is a very popular display technology, used for instance in television sets, monitors and mobile phones. This paper presents a new backlight dimming algorithm that exploits the characteristics of the target image, such as the local histograms and the average pixel intensity of each backlight segment, to reduce the power consumption of the backlight and enhance image quality. The local histogram of the pixels within each backlight segment is calculated and, based on this average, an adaptive quantile value is extracted. A classification into three classes based on the average luminance value is performed and, depending on the image luminance class, the extracted information on the local histogram determines the corresponding backlight value. The proposed method has been applied on two modeled screens: one with a high resolution direct-lit backlight, and the other screen with 16 edge-lit backlight segments placed in two columns and eight rows. We have compared the proposed algorithm against several known backlight dimming algorithms by simulations; and the results show that the proposed algorithm provides better trade-off between power consumption and image quality preservation than the other algorithms representing the state of the art among feature based backlight algorithms.

Keywords: Local backlight dimming, Liquid crystal display, Light emitting diode backlight, Image histogram, High dynamic range display

1. INTRODUCTION

Liquid Crystal Displays (LCDs) are very popular in television sets, monitors and projector systems. Since Liquid Crystals (LCs) do not emit light, LCDs require a light emitting *backlight* placed behind the LC pixels. Controlled with electrical signals, the LCs modulate the light and produce different colors with appropriate filters, usually Red, Green and Blue (RGB). Today, LEDs are commonly used as a light source for modern backlights. Traditional backlights were designed to illuminate the whole display uniformly. More recent backlight designs comprise several local segments or units that can be controlled independently; these are called *local* backlights, as opposed to traditional *global* backlights. Local backlights can be classified in *direct-lit* and *edge-lit,* according to the position of the light sources. In direct-lit backlights, the LEDs are placed behind the optical films and the LC panel, while in edge-lit backlights the LEDs are placed along the edge of the display. The light is distributed from the edges to the whole panel through a light guide [3, 6].

The use of several backlight segments allows *local backlight dimming*, which has the joint advantage of reducing power consumption and increasing the contrast ratio of the display. The principle of local backlight dimming is to adapt the backlight segments according to the image content, so that the backlight is darker behind the dark areas in the image. This clearly saves power, since the backlight does not always use full power. It also limits the light leaking through dark pixels (*leakage*), which consequently increases the contrast ratio. The reduction of light is compensated by adjusting the transmittance of the liquid crystals accordingly.

[1] ehsa@fotonik.dtu.dk; phone +45 4525 3890

Color Imaging XVIII: Displaying, Processing, Hardcopy, and Applications,
edited by Reiner Eschbach, Gabriel G. Marcu, Alessandro Rizzi, Proc. of SPIE Vol. 8652
86520V · © 2013 SPIE-IS&T · CCC code: 0277-786X/13/$18 · doi: 10.1117/12.2003119

Backlight dimming algorithms take a video frame as input signal, process it, and return a modified frame and a signal that drives the backlight. Figure 1 shows a frame and the backlights on direct-lit and edge-lit screens optimized for that frame.

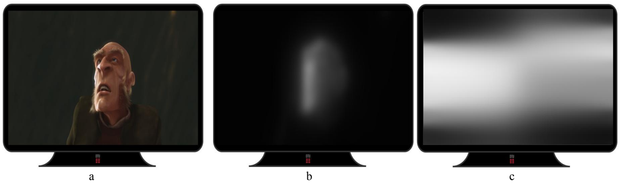


Fig. 1. Examples of different LCD backlights: a) test image; b) direct-lit backlight; c) edge-lit backlight.

In order to implement any efficient backlight dimming algorithm, it is necessary to model the essential characteristics of the display with local backlight dimming capabilities. Figure 2 illustrates a generic LCD with local backlight dimming. The observed luminance *l* of one LCD pixel can be calculated as the product of the backlight intensity *b* behind the pixel and the transmittance *t* of the pixels itself:

$$l = bt, \tag{1}$$

where *b* and *t* are normalized to the interval [0, 1]. In practice, *t* is never zero because of leakage, but this aspect can be modeled [1, 2]. Light from independent backlight segments is diffused and mixed on the diffuser plate between backlight units and LC layer. The diffusion of light from each LED follows a Point Spread Function (PSF) specific to the display and the LED, multiplied by the normalized luminance of the backlight unit. Each pixel position receives light from several LEDs, and the total luminance can be computed simply by summing the individual contributions. The backlight intensity at pixel (*i*,*j*) can be computed as follows:

$$b(i,j) = \sum_{k=1,N} r_k h_k(i,j), \tag{2}$$

where r_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 decreased the intensity significantly in the regions dominated by dark pixels, brightness compensation is performed to get the original target luminance. This is done by adjusting the LC transmittance *t* so that target *l* is reached if possible. More details are available in [2-4].

The backlight luminance can be determined by using characteristic data of the target image and the PSF, and several algorithms have been proposed to estimate it. The average, the square root of the average, and the maximum methods are the simplest algorithms [6]. They use fixed parameters derived from the target image to calculate the backlight intensities. These methods are simple but unreliable, as they do not always give a good trade-off between power savings and quality. In [7-12], different algorithms to derive the backlight luminance are introduced. These algorithms are combined with simple algorithms and the backlight luminance is defined by using the characteristics of the target image together with a simple algorithm. For example, algorithm in [7] uses the average value of the target image and a correction term to refine the backlight luminance.

More complex algorithms use global or local histograms of the target image pixel values [13-19]. Some algorithms use the information of the PSF to model light diffusion more accurately than others [1, 2, 5] which leads to more complex algorithms. In this paper, we present a backlight dimming algorithm that exploits the characteristics of the target image, such as local histograms and average pixel values of backlight segments, in order to reduce the power consumption of the backlights and enhance image quality for different types of images. We have also implemented many state-of-the art algorithms [5-12], and the performance of the proposed algorithm has been compared against them. The rest of the paper is organized as follows. Section 2 explains the backlight dimming algorithms known from the related studies. The proposed algorithm is presented in Section 3, and in Section 4 the simulation results are described. Finally, the concluding remarks are given in Section 5.

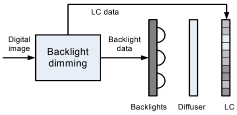


Fig. 2. Generic local backlight dimming system.

2. BACKLIGHT DIMMING ALGORITHMS

In order to compare different algorithms, we have selected some backlight dimming algorithms presented in the literature. The selected algorithms are described below.

The Max and Square-root algorithms [6] are simple methods, in which the intensity of each backlight segment is set to the maximum pixel value or the normalized square-root of the average pixel value in the respective segment. There are also simple methods that have been combined together with different strategies to introduce new methods [7-10]. For example, the algorithm proposed by Cho et al. [7] uses the relationship between the average and the maximum luminance in the input image to calculate the backlight luminance by adopting an additional correction term. The backlight luminance is calculated by:

$$r_k = mean(y_k) + corr, \tag{3}$$

$$corr = 0.5 \times \left(max(y_k) - mean(y_k) + \frac{(max(y_k) - mean(y_k))^2}{2^n} \right), \tag{4}$$

where r_k is the backlight luminance of segment k, y_k is a vector containing the luma pixels data in segment k, and n is the bit depth.

In addition, Zhang et al. [9] proposed a new method to reduce the clipping artifacts by increasing the backlight luminance from the average luminance. The backlight luminance is calculated by Eq. 3. Then, the values of the correction value *corr* are determined by measuring the loss of details when the backlight is reduced to the average pixel value of the input image and maximum pixel value of each block.

Kim et al. [10] proposed another algorithm to reduce light leakage and clipping in local dimming LCDs. The algorithm determines the backlight luminance of each block using a decision rule that derives the optimal value by considering the neighboring blocks and comparing light leakage and clipping measures.

Some algorithms use more features of the image, such as local or global histogram of the pixel values [13-19]. In [13], Lin et al. introduced a local dimming algorithm using the inverse mapping function. In the first step, a zone-weighting value of each backlight segment is decided, and then, the global histogram of the target image is computed to get a Probability Density Function (PDF). After that, the mapping function is calculated using the cumulative distribution function (CDF). Finally, inverting the mapping function (CDF) of the target image with the oblique line ($y = x$) generates a new mapping function for backlight modulation, named "Inverse of a Mapping Function (IMF)" [13].

Cho et al. introduced another local dimming algorithm which consists of several steps [14]. First, image histograms of each color in the local segments are generated, and the variance of luminance and contrast in the dimming images are calculated. Second, the backlight luminance of the local segments is determined based on the distortion of dimming images, and it is modified by comparing dimming level among nearby local segments to reduce the influence of light diffusion from other segments. In the end, the backlight luminance segments are calculated according to the dimming rates of the local segments.

Kang et al. [15] presented a multi-histogram-based algorithm for global dynamic backlight dimming. In Kang's algorithm, the input image is divided into blocks, each containing a backlight segment, to determine the multi-histogram. The multiple histograms are used to analyze the pixel distribution for RGB components of the input image. The maximum clipping point of the gray level (backlight luminance) is calculated using the total squared errors of the multiple histogram and maximum total square error of each block by using trial and error iteratively [15]. The total squared errors are calculated using the inverse of the target PSNR. Two methods, based on maximum value selection of the RGB and histogram binning are introduced for reducing the complexity of the algorithm [15]. S. I. Cho et al. [16] presented another version of local dimming using histogram analysis. In that method, the backlight luminance level of each segment is determined by performing two steps. In the first step, the algorithm dynamically selects the initial backlight luminance level of each segment, considering the peak signal-to-noise ratio and histogram analysis, as in Kang's algorithm [15]. In the second step, the backlight luminance is refined by considering the light coming from neighboring segments, as well as the relations between the current segment and its neighboring segments. In [17], histogram analysis has been used to define a clipping measure and reducing the clipping artifacts. In that algorithm, the backlight luminance is obtained iteratively.

The algorithm by Albrecht et al. [5] is 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 which 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.

3. PROPOSED ALGORITHM

The proposed algorithm uses the characteristics of the target image, including the histogram and average pixel value of each segment, to reduce power consumption of the backlight and to improve the image quality. Firstly, the target image is divided into segments following the shape of the backlight segments, and then the proposed algorithm is applied to each of the segments. The algorithm consists of three steps. First, the target image is classified in one of three classes. Then, the backlight luminance is initialized using histogram analysis. Finally, the resulting backlight luminance is calculated using the target image features.

3.1 Classification of the Target Image

In the first step, the average pixel value (including R, G, and B components for 8 bit pixels) of each backlight segment is first converted into physical domain by applying a standard Gamma function to the normalized pixel values and then re-quantizing the resulting values to eight bits. The resulting pixel values are used to map the target image into one of the three classes as follow:

$$class_k = \begin{cases} low & if \quad 0 \leq avg_k \leq 31 \\ medium & if \quad 32 \leq avg_k \leq 95 \\ high & if \quad 96 \leq avg_k \leq 255 \end{cases} \qquad (5)$$

where *k* is the backlight segment. The procedure to find the border values in Eq. 5 for classification was the following: first, each backlight segment of a target image (R, G, and B) was mapped into different classes by using the average algorithm. Then, each class was processed with various templates (about 12 templates) as shown in Fig. 3, and finally, the classification was done using the best results (Eq. 5) in terms of PSNR.

The procedure was performed using some challenging colorful, high contrast and low contrast images from different datasets (5 images from the Kodak [20] and 10 from the IEC [21] datasets).

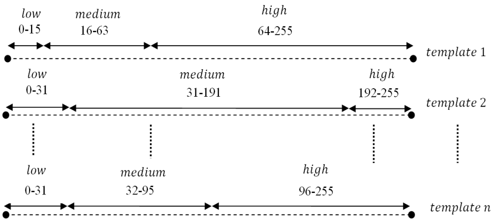


Fig. 3. Templates to find the best classes

3.2 Histogram Analysis for Initializing Backlight Luminance

After the backlight is dimmed, the pixel values are compensated to achieve the nearest value to the ideal target luminance. However, the maximum obtainable luminance is decreased to the level of the dimmed backlight luminance. In other words, after pixel compensation the displayed image and the target image need not to be the same, because the luminance of the displayed pixel cannot exceed the backlight luminance [4, 17]. Therefore, the luminance for high target intensity pixels cannot achieve the target luminance, even if the respective LC transmittance is enhanced to the maximum value of 255 (Fig. 4).

As shown in Fig. 4, the clipping occurs in high gray level values. In order to reduce the light leakage, a quantile analysis of the histogram is used to initialize the backlight luminance calculated locally for each backlight segment.

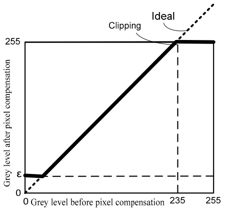


Fig.4. Clipping at high gray level when the pixel values are compensated (ε denotes light leakage [4])

The initial backlight level (IBL_k) is chosen from the histogram so that φ% of the pixel values are smaller than IBL_k (Fig. 5). φ is different for each class. A large IBL means that the image has many high intensity pixels and it can only obtain small power savings, whereas a small IBL indicates an input segment with many low intensity pixels, and thereby it is possible to achieve a better power saving ratio. In order to reduce the calculation load for the histogram, the gray level intensity of each pixel is defined as follow:

$$g = \max(R, G, B) \tag{6}$$

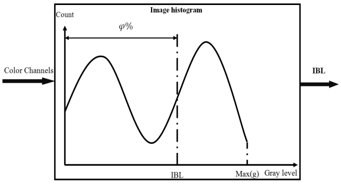


Fig. 5. Calculating the initial backlight value based on the histogram

3.3 Finding the Appropriate Backlight Luminance Level

The final step of the proposed algorithm is finding the appropriate backlight luminance level. In this step, the advantages of three intensity based backlight control algorithms are used [6]. The average value, the maximum value and the square-root of the normalized image pixel values in each segment are extracted. The average method could improve contrast ratio by decreasing luminance in dark scenes. The square-root method could enhance the backlight signal for scenes having intermediate brightness and the maximum algorithm could maintain the brightness and preserve the image details in bright scenes.

The luminance of the backlight is calculated by adding the IBL$_k$ at φ% of the maximum histogram and a feature value:

$$r_k = IBL_k(\varphi) + Z_k \times feature_k \, , \; Z_k \in \{\alpha, \beta, \gamma\}, \qquad (7)$$

where the *feature* is a function of the average, the square-root and the coefficient of variation of each target image segment (R, G, and B). The square-root is the square-root of the average normalized pixel value. The features are controlled by parameters $\alpha, \beta,$ and γ, determined via trial and error. The optimal values of these parameters are determined by the tradeoff between the quality and the power saving obtained when different values were applied to 10 images out of the set depicted in Fig. 7 and some challenging images from different IEC [21] test set, like high contrast, colorful, high and low luminance images. In this paper, $\alpha, \beta,$ and γ are set to 0.2, 0.5, and 0.8, respectively. The initial backlight level (IBL) is defined at the φ = 60, 65 and 70 quantile the target image for the low, medium and high brightness classes, respectively.

Figure 6 shows the proposed algorithm for determining the luminance level of the backlight. The *diff* in Fig. 6 is the difference between the maximum and the average value of g_k (6). The coefficient of variation (cv_k) is calculated using by the average and the variance values for each segment.

$$cv_k = \frac{var(g_k)}{mean(g_k)} \qquad (8)$$

where $g_k$is given by (6). The correction term cv_k is used to increase the backlight level in the brightest class, because otherwise the method would produce too low backlight intensity for the smooth and bright segments.

4. EXPERIMENTS AND SIMULATION RESULTS

To evaluate the performance of the proposed algorithm against other backlight algorithms, objective and subjective experiments have been conducted. The objective evaluation compares the results of the proposed version with 11 algorithms from the literature over 32 images while, for practical reasons, the subjective evaluation compares it to 4 of the other algorithms on 7 images. The two evaluation processes are described in Sections 4.1 and 4.2, respectively. All the algorithms have been implemented in Matlab.

```
Partition the image corresponding to the backlight segments.
  I.     Classification of each target image segment in one of three (Low, medium, high) classes using average
    value ($avg_k$) of R, G, and B using (5).
  II.    Finding the backlight luminance ($r_k$) as follows:
         $if\ class = Low\ do$
                 $r_k = P(IBL_k(60), avg_k);$
         $else\ if\ class = Medium\ do$
                 $r_k = P(IBL_k(65), sqrt_k);$
         $else\ class = High\ do$
                 $r_k = P(IBL_k(70), (sqrt + cv)_k);$
         $end$
The function $P$ is calculated as follow for each segment:
$P(IBL, x)$
         $if\ 0 \le diff \le 31$
              $r = IBL + \alpha \times x$
         $elseif\ 32 \le diff \le 95$
              $r = IBL + \beta \times x$
         $if\ 96 \le diff \le 255$
              $r = IBL + \gamma \times x$
         $end$
```

Fig. 6. Proposed algorithm for controlling the backlight luminance

4.1 Objective evaluations

The simulations have been run on 32 different images (Fig. 7). Of these, 24 belong to the Kodak True Color Image Suite [20]. 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 [3, 4]. All images have been rescaled to Full HD resolution with Bicubic interpolation (1980x1080). For the simulation, the input RGB images have been normalized with the standard Gamma function, γ=2.2 first. The resulting backlight intensity values have been calculated in physical domain as an 8 bit depth image (Fig. 6). Then, backlight luminance values have been normalized between 0 and 1 for calculating the relative power. Finally, the inverse Gamma function has been applied to the normalized result to return to a perceptually uniform representation. The reported MSE results are based on the normalized values in perceptual domain.

Two screen models have been used for the experiments. The first one is based on an edge-lit display with 16 backlight segments placed in eight rows and two columns. 16 individually measured PSF have been used. The second is based on a 47" local backlight dimming Full HD screen from SIM2 [22]; this screen has 2202 direct-lit backlight segments placed in a hexagonal grid. One PSF has been used for all segments. Two different leakage factors ε (0.001 and 0.0002) have been used in all simulations. After applying dimming algorithms, hard clipping in (i.e. pixels that cannot be fully satisfied are enhanced to the maximum value, see Fig. 4) pixel compensation was performed, as described in [4].

The power consumption of the backlight is calculated using the average of LED values (2). Mean Squared Error (MSE) and Peak Signal-to-Noise Ratio (PSNR) have been used to evaluate the resulting image quality. The results are averaged over the 32 images. The simulated results can be compared against the target image using numerical quality indicators, in this case PSNR and MSE.

Fig. 7. The test set images used in the experiments

Some backlight dimming algorithms have been selected to compare our approach against other algorithms. These algorithms are: the SQRT (square root of the average pixel value used for the backlight), the Max (maximum pixel value used for the backlight) presented in [6], Cho et al. [7], Wang et al. [8], Zhang et al. [9], Kim et al. [10], Lin et al. [13], Cho et al. [14], Kang et al. [15], S. I. Cho et al. [16] and Albrecht et al. [5]. For brevity, we have named these algorithms according to the first author. For the Albrecht algorithm, the normalized RGB values have been linearized with the standard Gamma function and in addition, two of the three steps have been implemented, which produces a clipper-free result. There are two Cho algorithms, in this paper named Cho [7] and Cho13 [14] for the first and the second algorithm, respectively. Since Cho13 is a color local dimming algorithm, and in our implementation the color backlight is not used, the target image is converted to a gray level image. Kang's algorithm is a global dimming algorithm, and it has been generalized to local dimming. In the generalized form, the number of blocks in each backlight segment has been set to four [15]. Three different versions of Kang algorithms have been implemented, the first one is the global algorithm (global Kang), the second and the third are the low target PSNR (set to 30 db) denoted as Kang Low, and the high target PSNR (set to 50 db) denoted as Kang High, respectively. All the algorithms are applied to both SIM2 and edge-lit displays, except S.I. Cho, that was only applied to the edge-lit display.

Figures 8 and 10 show plots comparing the MSE of the proposed algorithm and the other algorithms on both edge-lit and direct-lit backlights, using two leakage values (0.001, 0.0002). As shown in Fig. 8, the proposed algorithm has the best performance in terms of quality, when the leakage is 0.001. In the other cases, better trade-off between power consumption and image quality can be achieved with the proposed algorithm, compared to the other algorithms.

To use the LED and LC values on real screens, they need to be quantized. For the backlight intensity, quantization has a negligible impact, but the LC quantizing has a significant impact on the numeric quality [4]. In Fig. 9, the results for LC values quantized to 8 bit are given. As a result, the SQRT, Kang's algorithm (low), Kim's algorithm, Cho's algorithms reduce power consumption of backlight, but the resulting image quality is worse than with the other algorithms. With the Max, Zhang's, Kang's algorithm (high), Albrecht's, S.I. Cho's algorithm, the new Cho's algorithm (Cho13), Wang's and Lin's algorithms, image quality is good. However, the backlight power saving is small. With the proposed algorithm, image quality and backlight power savings for the various images are better balanced compared with the others. As shown in the figures, the backlight algorithms could be divided in two different categories: low power algorithms and high quality algorithms. The proposed method provides a good compromise between image quality and power consumption.

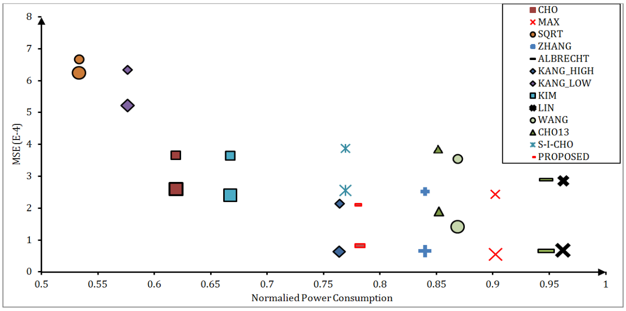


Fig. 8. Distortion (MSE) vs. power trade-off of the proposed algorithm, compared with other algorithms on edge-lit backlight. (The smaller points ε=0.001, and the bigger points ε=0.0002).

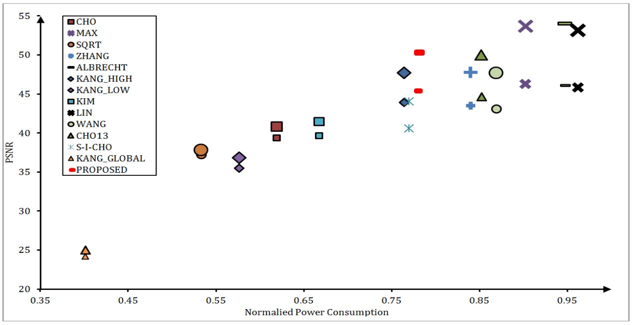


Fig. 9. PSNR vs. power trade-off of the proposed algorithm compared with other algorithms on edge-lit backlight for 8 bit LC quantization. (The smaller points ε=0.001, and the bigger points ε=0.0002).

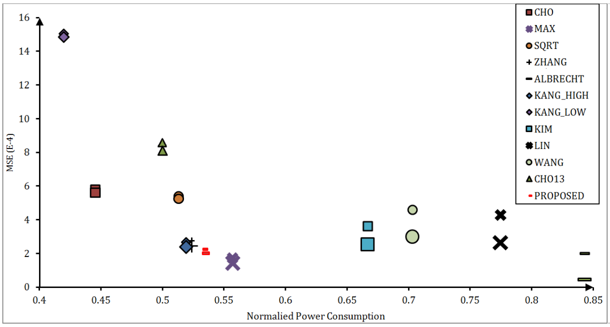


Fig. 10 Distortion (MSE) vs. power trade-off of the proposed algorithm compared with other algorithms on direct-lit backlight. (The smaller points ε=0.001, and the bigger points ε=0.0002). Because the Kang's algorithm (Global) has far result, it has been removed.

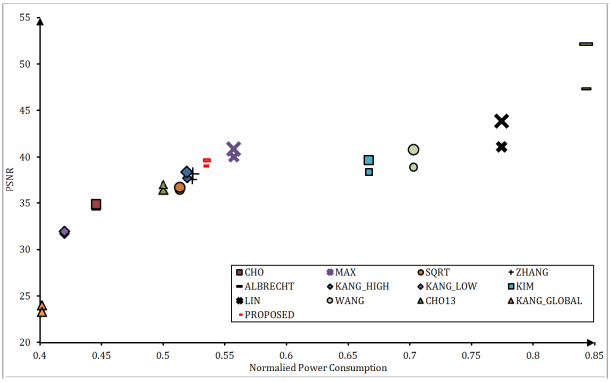


Fig. 11 PSNR vs. power trade-off of the proposed algorithm compared with other algorithms on direct-lit backlight for 8 bit LC quantization. (The smaller points ε=0.001, and the bigger points ε=0.0002).

4.2 Subjective Evaluation

In order to evaluate performance of the backlight algorithms, we set up a subjective paired comparison experiment. The experiment was designed to respect the rules presented by the ITU in [23]. To limit the duration of the experiment, we chose a subset of 7 images (2 Kodak images [20], Volcano and Diver images [24], and Exotic flower and Lizard images [25], and Stars image [4]), and 6 backlight dimming algorithms to be compared. In the test, the observers were asked to choose the one they prefer between two versions of an image. They could navigate between those two versions for as long as they wish before selecting the version they prefer. The participants were first given instructions; they then performed 4 training trials to get familiar with the methodology before doing the actual experiment. The images used during the trials were not part of the experiment dataset. Images were displayed by simulating the 16 segment edge-lit on the SIM2 screen with the observer standing at a distance of 1,76 m, that equals three times the height of the display (58,53 cm) from the observer. Participants were sitting on a chair right in front of the screen in order to ensure a perpendicular view to the screen. A border of 2 cm at the edges of the screen is hidden to avoid reflection issues. The experiments were done with lights off and curtains closed. The images used for the experiment are depicted in Fig. 12.

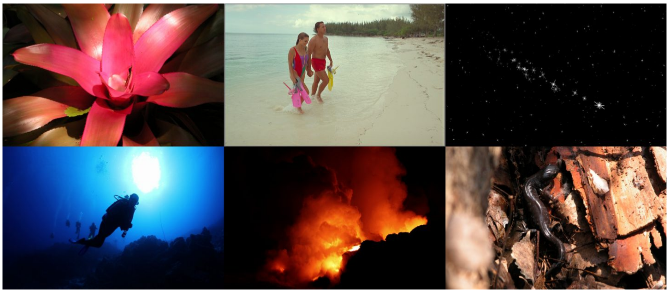


Fig. 12. Images used for the subjective experiment (seventh image is the parrot image in Fig.7)

The experiment was done by 16 observers who were all naive regarding the purpose of the test and non-experts in backlight dimming. Their age ranged from 22 to 30 and they were mostly men (12 men, 4 women). One test session lasted between 10 and 16 minutes and each participant performed the test two times, so the complete duration of the experiment was around 30 minutes (never more than 35min).

The obtained subjective comparisons were transformed into absolute ratings using the Thurstone-Mosteller model described in [26, 27] and the maximum a posteriori model to deal with unanimous ratings (i.e. when all or almost all participants prefer one version to the other), as described in [28]. Table 1 shows the obtained ratings scaled to the same range: the higher the subjective grade is, the more participants preferred this version.

We can see that the proposed version most of the time ranks in second place, after Albrecht and before Zhang (it is always in top three). However, with a few exceptions, the power consumption for Albrecht and Zhang algorithms is also higher than for the proposed algorithm. In addition, Albrecht's algorithm is an iterative and complex algorithm, while, the proposed algorithm is non-iterative and thus a simpler algorithm. The quality ranking based on subjective scores follows reasonably well the quality ranking based on PSNR values. Therefore, we suggest that PSNR can be used as a tool for approximating the relative visual quality in simulated backlight dimming systems. A notable exception is the Lizard image, on which the proposed algorithm outperforms Zhang's by subjective evaluation, even though PSNR is lower. Visual inspection reveals that there is a clipping area using Zhang's algorithm in an attention region that could not be noticed for the proposed algorithm, that was noticeable for the subjects. We assume that the clipped pixels with the proposed algorithm are distributed more evenly in the low attention areas and therefore less noticeable, even though PSNR is lower.

Table 1. Subjective and PSNR ratings for the experiment stimuli, compared against power consumption

Image Name	Grade Type	Proposed	Zhang [9]	Albrecht [5]	Kang Low	Cho [7]
Stars	Subjective	0.56	0.58[2]	0.58	0.52	0
	PSNR	35.4	33.76	33.76	35.83	35.69
	Power	0.59	0.98	0.98	0.50	0.49
Beach	Subjective	0.86	0.73	0.94	0	0.76
	PSNR	57.06	53.54	57.34	26.07	48.15
	Power	0.89	0.80	0.85	0.45	0.77
Parrot	Subjective	0.99	0.84	0.88	0	0.45
	PSNR	53.97	53.55	54.35	28.08	42.88
	Power	0.87	0.81	0.84	0.45	0.66
Volcano	Subjective	0.55	0.73	0.76	0.14	0
	PSNR	34.98	37.06	37.83	28.14	30.65
	Power	0.54	0.77	0.66	0.38	0.44
Exotic Flower	Subjective	0.27	0.52	0.85	0.18	0
	PSNR	37.61	41.88	45.42	34.84	28.26
	Power	0.81	0.93	0.98	0.75	0.63
Lizard	Subjective	0.67	0.50	0.87	0	0.05
	PSNR	40.19	45.13	45.15	32.72	33.04
	Power	0.75	0.91	0.97	0.55	0.59
Diver	Subjective	0.45	0.37	0.51	Excluded[3]	0
	PSNR	44.93	37.02	45.85	26.08	30.65
	Power	0.70	0.74	0.82	0.47	0.56
Average	Rank-Subjective	2.28	2.43	1.28	4.57	4.43
	PSNR(Ranking)	43.45(2.57)	43.13(2.71)	45,67(1.57)	30.25(4.29)	35.62(3.86)
	Power	0.74	0.85	0.87	0.51	0.59

CONCLUSIONS

In this paper, we present a low complexity and adaptive algorithm for local dimming of LED backlight displays. The proposed algorithm uses the local characteristics of the target image, such as the local histograms and the average pixel intensity, to find the backlight luminance. The proposed method has been applied on two modeled screens using a large selection of different types of images, for instance high-contrast and high-luminance images. The first modeled screen has a high resolution direct-lit backlight, while the other screen has edge-lit backlight with 16 segments placed in two columns and eight rows. Results show that the proposed algorithm can achieve better trade-off between power consumption and image quality, when compared against other feature based algorithms known from the literature.

[2] For this image, the LED computed with Zhang's algorithm are almost identical to those computed with Albrecht's version (the difference is <0,001). Only one of them was shown to the participants.

[3] We excluded this combination from the experiment to keep its duration within an acceptable range. The quality of image was low.

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] Shu, X., Wu, X., and Forchhammer, S., “Optimal local dimming for LC image formation with controllable backlighting, ” IEEE Trans. Image Proc. DOI 10.1109/TIP.2012.2211371, (2012).

[3] Burini, N, Nadernejad. E, Korhonen. J, Forchhammer. S,” Speedup of optimization-based approach to local backlight dimming of HDR displays,” Proc. of SPIE Vol. 8436, 84360B (2012).

[4] 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).

[5] 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).

[6] Seetzen, H., Heidrich, W., Stuerzlinger, W., Ward, G., Whitehead, L., Trantacoste, M., Ghosh, A., and A. Vorozcovs, “High dynamic range display systems,” ACM Transactions on Graphics, vol. 23, no. 3, p.760, Aug. 2004.

[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, vol. 55, 839–844 (2009).

[8] Wang, G-Z., Lin, F-C., and Huang, Y-P., “Delta-Color Adjustment (DCA) for Spatial Modulated Color Backlight Algorithm on High Dynamic Range CD TVs ,” Journal of Display Technology,vol. 6, no. 6, pp. 215-220 , June 2010.

[9] Zhang, X.-b., Wang. R., Dong. D., Han. J.-h., and Wu. H.-x.,” Dynamic Backlight Adaptation Based on the Details of Image for Liquid Crystal Displays,” Journal of Display Technology, vol. 8, no. 2, pp. 108–111, Feb. 2012.

[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 display, ” Journal of the Society for Information Display, vol. 17, no. 12, p. 1051, 2009.

[11] Nam, H., “Low power active dimming liquid crystal display with high resolution backlight,” Electronics Letters, vol. 47, no. 9, 538-540, (2011).

[12] Chen, C., Sung, J., Ha, T., Park, Y.,” Locally pixel-compensated backlight dimming on LED-backlit LCD TV,” Journal of the SID, 15(12), 981-988 (2007).

[13] Lin, F.-C., Huang. Y.-P, Liao. L.-Y, Liao. C.-Y, Shieh., H.-P.D, Wang. T.M, and Yeh. S.-C., ” Dynamic Backlight Gamma on High Dynamic Range LCD TVs ,” Journal Display Technology, vol. 4, no. 2, pp. 139-146, 2008.

[14] Cho, H., Cho, H.C., Hong, H.J., Oh, E.-Y., and Kwon, O. K., “A color local dimming algorithm for liquid crystals displays using color light emitting diode backlight systems,” Optics & Laser Technology, vol. 47, 80–87 (2013).

[15] Kang, S.-J, and Kim. Y. H, ” Dynamic Backlight Gamma on High Dynamic Range LCD TVs,” Journal Display Technology, vol. 7, no. 10, pp. 544-549, 2011.

[16] Cho, S. I., Kim, H.-S., Kim, Y. H., “Two-step local dimming for image quality preservation in LCD displays,” IEEE Int. SoC Design. Conf. (ISOCC’11), (2011).

[17] Hong, J.-J., Kim, S.-E., Song, W. J., “A clipping reduction algorithm using backlight luminance compensation for local dimming liquid crystal displays,” IEEE Trans. on Consumer Electronics, vol. 56,no. 1, pp. 240–246, Feb. 2010.

[18] Lai, C.-C., Tsai, C.-C., “Backlight power reduction and image contrast enhancement using adaptive dimming for global backlight applications, ” IEEE Trans. on Consumer Electronics, vol.54, no.2, pp. 669-674, May. 2008.

[19] Jung, H., Lee, Y., Suh, D.Y., “Cooperative local dimming for accurate backlight brightness matching,” Electronics Letters, vol. 47, no. 4, 252-256, (2011)

[20] Kodak True Color Image Suite, <http://r0k.us/graphics/kodak/>.

[21] “Method of Measurement for the power Consumption of Audio, Video and Related Equipment,” IEC 62087 Ed. 2.0, 2008.

[22] SIM2 HDR TV, <http://www.sim2.com/HDR/>.

[23] Recommendation ITU-R BT.500-12, “Methodology for the subjective assessment of the quality of television pictures”, 2009.

[24] The Consumer Digital Video Library, <http://www.cdvl.org>.

[25] Olmos, A., Kingdom, F. A. A, “ A biologically inspired algorithm for the recovery of shading and reflectance images”, *Perception*, 33, 1463 - 1473, (2004).

[26] Handley, J. C., "Comparative analysis of Bradley-Terry and Thurstone-Mosteller paired comparison models for image quality assessment", Im. Proc., Im. Qual., Im. Capture Syst. Conf. (PICS'01) pp. 108-112, 2001.

[27] Woods, Russell L.; Satgunam, Premnandhini; Bronstad, P. Matthew; Peli, Eli, “Statistical analysis of subjective preferences for video enhancement”, Proceedings of the SPIE Human Vision and Electronic Imaging XV, Volume 7527, 2010.

[28] Tsukida, K. and Gupta, M. R., " How to Analyze Paired Comparison Data", UWEE Technical Report, 2011.

A.5 Flicker Reduction in Led-Lcds with Local Backlight

E. Nadernejad, C. Mantel, N. Burini, and S. Forchhammer, “Flicker Reduction in LED-LCDs with Local Backlight”, in *2013 IEEE 15th International Workshop on Multimedia Signal Processing (MMSP)*, IEEE, 2013

Reference: [5]

Flicker Reduction in LED-LCDs with Local Backlight

Ehsan Nadernejad, Claire Mantel, Nino Burini and Søren Forchhammer

Department of Photonics Engineering, Technical University of Denmark
2800 Kgs. Lyngby, Denmark
{ehsa,clma,nibu,sofo}@fotonik.dtu.dk

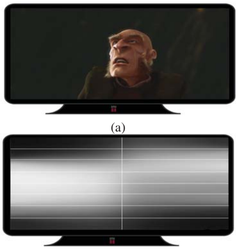
***Abstract*—Local backlight dimming of LCD with LED backlight can reduce power consumption and improve quality of displayed images and videos. However, important variations of LED over time produce a visually annoying artifact called flickering. In this work, we propose a new algorithm to reduce flickering while maintaining video quality. The proposed algorithm uses an adaptive second order Infinite Impulse Response (IIR) in which coefficients are calculated from the local image features. Experimental results show that the proposed method can reduce flickering while simultaneously keeping similar video quality in terms of PSNR and MSE.**

I. Introduction

Nowadays, Liquid Crystal Displays (LCD) are used in different systems, like TV sets, computer monitors and etc. They have high dynamic range, high contrast ratio and low power consumption. Images are rendered by Liquid Crystal (LC) pixels filtering the light that comes from the backlight, located behind them [1], [2]. When the light source is placed behind the LC layer, the backlight is called direct-lit, otherwise it is referred to as an edge-lit backlight [3]. The backlight dimming algorithm controls the backlight luminance according to target image, dynamically. Conventional LCDs utilize only one backlight segment covering the whole display, defining global backlights. Global backlight dimming methods consume much power and cause light leakage in the black areas [3]. To overcome this, local backlights were introduced. Local backlight can be dimmed adaptively to match the image content, reducing power consumption and increasing contrast. Figure 1 shows an example of a local backlight dimming algorithm on an edge-lit display.

Several algorithms for backlight dimming have been proposed. Some algorithms use characteristic data of the target image [1], [4], [5], while more complex use the information of the Point Spread Function (PSF) to model light diffusion [2], [3], [6], [7]. By using these algorithms, LCDs achieves high contrast ratio and low power consumption.

A rendered video frame is formed by multiplying the light emitted by the backlight with the transmittance of the LC pixels [1]. In video sequences, when the image content changes or moves very quickly and frequently, the calculated backlight luminance could also be changed very rapidly and frequently as shown in Fig. 2. This phenomenon creates an unnatural trembling in the backlight segment, which defines the flickering artifact and is caused by temporal local or global dimming.


(a)

(b)

Fig. 1. Example of a local backlight dimming, a) Image frame, b) Locally dimmed backlight luminance on edge-lit display.

Some algorithms for flicker removal were proposed in the literature [8], [9]. The first algorithm uses Moving Average (MA) filter taking the average pixel values through several consecutive frames. Since, this algorithm is simply implemented, it has been generally applied to the local dimming LCDs. However, the MA filtering can not be generalized to the other dimming algorithms like PSF-based or histogram-based algorithms because it uses average luminance value of the target image for removing flickering artifact while most of the backlight dimming algorithm use other features (PSF or histogram) [1], [4], [5]. In addition, it causes a delay in the backlight when a scene change occurs. In [8] an adaptive MA filtering algorithm was proposed to remove flicker. This algorithm uses maximum 16 previous frames adaptively according to each backlight segment [8]. In Chen et. al [9], an Infinite Impulse Response (IIR) filter were proposed to smooth temporal backlight variation and eliminate flickering. This algorithm uses temporal and spatial backlight luminance to avoid flickering artifact [9]. While most flicker removal

MMSP'13, Sept. 30 - Oct. 2, 2013, Pula (Sardinia), Italy.
978-1-4799-0125-8/13/$31.00 ©2013 IEEE.

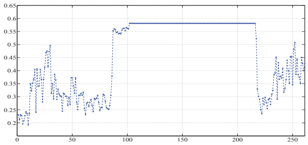


Fig. 2. Example of the flickering artifact in LED segment No.4 of the edge-lit display (The video is *'Stars'* [10] on Cho's algorithm [5] with 261 frames).

methods are designed for a specific backlight algorithm, they can not be generalized to be used in other algorithms. The average values of the target image are considered as a feature in most flicker removal algorithms.

Chen et. al [9] proposed a first order IIR filter for flicker removal in which the filter coefficients are calculated using average values of the target image. In this paper, we propose an adaptive flicker-reduction algorithm for LED-LCDs with dynamic backlight. The proposed algorithm is a second order IIR filter which uses information of the two previous spatial and temporal frames to calculate the backlight luminance. The filter coefficients are calculated adaptively using local features of the target image. The difference between maximum and average luminance values of the target image in each backlight segment is used to calculate the filter coefficients. The experimental results show that the proposed algorithm has a good performance in flicker removal. For some video sequences with very high flicker level, the algorithm can be repeated two or three times.

The rest of this paper is organized as follows. The proposed algorithm is described in Section II. Section III shows the experimental results on different video sequences and different local dimming algorithms. Finally, we conclude in Section IV.

II. Flicker Reduction Using IIR Filter

An annoying artifact which may be caused by local dimming is screen flicker. When the frame content of a video sequence change or move very quickly and frequently, the calculated LED luminous intensities may also change likewise. Rapid backlight can cause flickering. When backlight luminance intensity changes more than pixel values do, the flicker artifact will be more obvious. Therefore, a temporal filter is necessary to reduce the flicker artifact by temporally smoothing the change in the backlight. Figure 2 shows an example of the flickering artifact on an edge-lit display when the Cho algorithm [5] is applied to a video sequence.

An IIR filter can reduce the flickering artifact. One problem with using a smooth IIR filter is that the change in the backlight lags behind the change of the image signal. This artifact is more obvious at the moment of scene change, especially from a bright scene to a black scene. To reduce and remove screen flicker and backlighting lag artifacts an adaptive second order IIR is proposed. Figure 3 illustrates the block diagram of the proposed IIR filter.

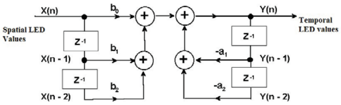


Fig. 3. Block diagram of the proposed IIR for flickering reduction. (Z^{-1} is the delay operator)

The general form of the second order IIR filter in differential form is as follows:

$$y[n] = a_1 y[n-1] + a_2 y[n-2] + b_0 x[n] + b_1 x[n-1] + b_2 x[n-2] \quad (1)$$

where $y[n]$ and $x[n]$ are output and input of the filter. In order to calculate the temporal output of the backlight luminance, a linear combination of the spatial and the temporal variables is used as shown in Eq.1. The temporal backlight luminance is as follows:

$$L_T^k = (a_1 L_T^{k-1} + a_2 L_T^{k-2} + b_0 L_S^k + b_1 L_S^{k-1} + b_2 L_S^{k-2})/\lambda \quad (2)$$

where L_T^k and L_S^k are the temporal and the spatial LED values that correspond to $y[n]$ and $x[n]$ of the IIR filter, k, $k-1$ and $k-2$ indicate the current and the last two frames, and a_1, a_2, b_0, b_1 and b_2 are the parameters controlling the smoothness of the IIR low-pass filter. The coefficients are selected adaptively according to the video content and are calculated as

$$\begin{aligned}
&b_2 = 0;\\
&b_0 = \min(1, TH_1 + |diff_m^k - diff_m^{k-1}|;\\
&b_1 = \min(1, TH_2 + |diff_m^{k-1} - diff_m^{k-2}|; \qquad (3)\\
&a_1 = 1 - b_0;\\
&a_2 = 1 - b_1;\\
&\textit{and}\\
&\lambda = a_1 + a_2 + b_0 + b_1 + b_2;
\end{aligned}$$

where TH_1 and TH_2 are predefined parameters used to control the shape of IIR filter, k is the frame number, m is the backlight segment number, and $diff_m^k$ is the difference between the average and the maximum of the normalized pixel values in the entire segment. It is calculated as follows:

$$diff_m = max_m - avg_m \quad (4)$$

where max_m and avg_m are the maximum and the average values of the normalized pixel values in physical domain. A larger difference between $diff^k$, $diff^{k-1}$ and $diff^{k-2}$ increases the possibility of scene change, so larger coefficient values are needed. This means that the temporal output is not smooth enough. Using the adaptive IIR filter, the changes in

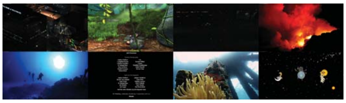


Fig. 4. A snapshot of each test video sequences from [12]. The name of sequences from left to right and top to bottom are: *Theatre*, *BBBunny* [12], *Concert*, *Volcano* [12], *Diver* [12], *Titles* [10], *Anemone* [12], and *Stars* [10], respectively.

the backlight output increases when there are changes in the scene, otherwise the backlight changes smoothly. In this way, both flicker and backlight lag are reduced.

III. Experiments

The performance of the proposed algorithm was tested on some state-of-the-art backlight dimming algorithms. The algorithms were Albrecht's algorithm [6], Cho's algorithm [5], Zhang's algorithm [4], Nadernejad's algorithm [1] and the optimization algorithm [2], [11]. For Albrechts algorithm [6], we have implemented the first two steps (out of three, the third one being optional), which produces a clipper-free result. Two versions of the optimization algorithm were implemented [11]. These are based on Gradient Descent (GD) optimization algorithm with two leakage factors (GD with $\epsilon = 0.001$, and GD with $\epsilon = 0.0002$ [11]). All algorithms were simulated on an edge-lit display with 16 backlight segments placed in eight rows and two columns.

To assess the performance of our algorithm on video sequences, we tested them on 8 Full-HD (1920x1080) sequences. Four sequences (*Volcano*, *Anemone*, *BBBunny* and *Diver*) come from [12], two (*Stars* and *Titles*) come from [10] and two come from DVD content. These sequences present varying characteristics in terms of high and low luminance, color and details, as well as temporal variation (motion and scene change). Figure 4 shows a frame extracted from each of them. In the experiments $TH_1 = TH_2 = 0.125$.

Figure 5 shows the result of the proposed algorithm on different video sequences for segment number 13 of edge-lit display using Cho's algorithm [5]. As can be seen, when the level of flicker is low like the *Volcano* and *Concert* sequences, the algorithm preserves the LED values. However, in the case of high flicker like *Stars* and *Theatre*, the fluctuations in LED values are controlled in an acceptable level. The control level follows the normal trend of the previous LED values. The effect of the proposed algorithm is the same for other segments. In Fig. 6, the average LED values (The average for all 16 segments of the edge-lit display) for one video sequence '*Stars*' are shown. Another advantage of this method is that the average power consumption does not change significantly after flicker removal. The power level before and after filtering are shown at the bottom of Fig. 6.

In order to evaluate the performance of the proposed algorithm based on the objective quality measures the PSNR and MSE vectors for *Stars* video sequence after and before filtering are illustrated in Figs. 7 and 8 (The LC values are quantized to 10 bit). As shown in the these figures, the proposed algorithm can produce a flicker-free result while keeping the quality of the video signal. Although, the PSNR and MSE after filtering were not always better compared to those of the original video sequence, their averages were almost the same as shown in the figures. In addition, their overall fluctuations were smoothed.

The average results for different local dimming algorithms on different sequences are summarized in Table I. These results are for the *Title* (251 frames), *Theatre* (207 frames), *Stars* (255 frames), *Diver* (300 frames), *Concert* (217 frames), *Anemone* (300 frames), *Volcano* (300 frames), and *BBBunny* (250 frames) video sequences. It is important to keep the quality of the videos at their initial level before filtering. Therefore, in this table, the minimum as well as the average PSNR are reported and due to its reverse relationship with the MSE, the corresponding maximum and average MSEs are shown. For power consumption, all the minimum, maximum and average values are reported. As observed in the table, the proposed filter does not have a large impact on PSNR, MSE and the power consumption, in terms of minimum, maximum and average values. In addition, Figure 9 shows the average quality versus power consumption of the different algorithms applied on all the sequences before and after filtering. In general, a flicker removal algorithm should not affect the quality and power consumption of LEDs. These results verify that the proposed algorithm preserves these two characteristics since the position of the points in the plot are close to each other before and after filtering. There is no objective metric for the flickering artifact in LCD displays with local backlight. However, we evaluated the performance of the filter both visually and objectively. The temporal variation is computed by the SSD as recommended in [13] to measure flickering caused by video compression. SSD is defined as:

$$SSD = \frac{1}{N_{frames}} \sum_{n} \frac{1}{N_{pixels}} \sum_{i} \left(Fr_{n-1}(i) - Fr_n(i)\right)^2 \tag{5}$$

where $Fr_n(i)$ is the pixel i of the frame n. The SSD results for all sequences are shown in Table II. The average level of temporal variations is either smaller after temporal filtering (TF) than before applying the filter $(Orgn)$ or similar. In some cases that the level of the flicker was quite high, the algorithm was applied two or three times for a successful removal. This is shown by the numbers inside the parentheses in Table II. In some cases, the removal was not completely successful and this is shown by $(*)$ in the table that means not removed completely.

IV. Conclusion

In this paper an adaptive second order IIR filter for removing flickering artifact in LED-LCD with local backlight was proposed. The filter uses spatial and temporal information of LED values. The coefficients of the IIR were selected adaptively using local features of the target image. To show

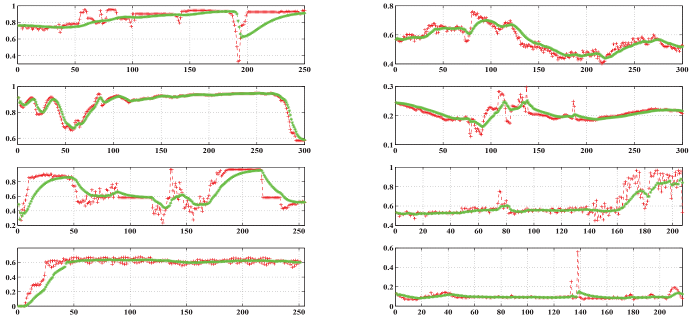


Fig. 5. Performance of the proposed algorithm on different sequences, visualized using an edge-lit display in which the Cho's algorithm [5] is used. The red and green line show the LED values before and after filtering, respectively. (LED No.13, Column. 1: 1,...4 are: *BBBunny*, *Volcano*, *Stars*, *Title*, and column. 2: 1,...4 are: *Anemone*, *Diver*, *Theatre*, *Concert*.)

TABLE I

THE AVERAGE DE-FLICKERING RESULTS FOR DIFFERENT ALGORITHMS ON ALL VIDEO SEQUENCES. THE ALGORITHMS ARE: 1: GD WITH $\epsilon = 0.0002$ [2], 2: ALBRECHT [6], 3: CHO [5], 4: NADERNEJAD [1], 5: ZHANG [4], AND 6: GD WITH $\epsilon = 0.001$ [2], RESPECTIVELY.

		Before						After					
Metric		1	2	3	4	5	6	1	2	3	4	5	6
	Mean	0.824	2.737	2.385	3.769	1.765	1.108	0.957	2.639	2.410	3.927	1.769	1.522
MSE (E-4)	Max	1.130	5.787	4.093	9.316	4.877	1.526	1.455	5.078	3.935	7.530	3.385	2.641
	Mean	47.537	42.090	39.878	38.304	41.419	45.092	47.005	42.595	39.680	38.024	41.349	44.285
PSNR	Min	44.898	37.354	37.393	35.129	38.150	42.778	43.458	37.689	36.108	34.580	37.544	39.927
	Mean	0.660	0.675	0.566	0.471	0.607	0.531	0.668	0.697	0.572	0.476	0.613	0.525
	Min	0.480	0.492	0.433	0.377	0.470	0.377	0.538	0.597	0.479	0.409	0.520	0.401
Power	Max	0.782	0.828	0.673	0.548	0.710	0.646	0.724	0.653	0.596	0.495	0.636	0.624

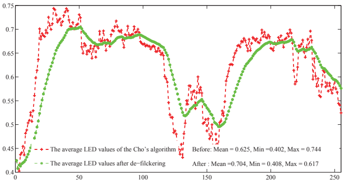


Fig. 6. The average led values for 16 backlight segments before and after filtering applied on *Stars* using Cho's algorithm [5].

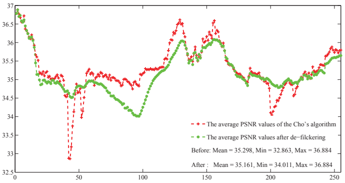


Fig. 7. PSNR of the *Stars* sequences before and after de-flickering.

the performance of the proposed algorithm, some state-of-the-art local backlight dimming algorithms were implemented and tested on 8 Full-HD video sequences with different features. Experimental results demonstrated that the proposed algorithm can avoid abrupt variations of the LED signals and consequently reduce the flicker artifact.

ACKNOWLEDGMENT

We would like to thank Jesper Meldgaard Pedersen, Bang & Olufsen for discussions and contributions.

TABLE II
SSD measures of sequences before and after temporal filtering for different algorithms on all video sequences. (The numbers inside the parentheses show the number of times the filtering was repeated and (∗) means that the flicker was not removed completely.)

		GD $\epsilon = 0.0002$ [2]		Albrecht [6]		Cho [5]		Nadernejad [1]		Zhang [4]		GD $\epsilon = 0.001$ [2]	
		Orgn	TF	Orgn	TF	Orgn	TF	Orgn	TF	Orgn	TF	Orgn	TF
Stars	Avg	64,83	59,79	67,34	66,43	65,37	64,04	55,79	52,87	65,24	63,72	59,66	52,37
	Max	137,87	122,78	140,73	138,74	133,88	134,24	115,24	112,01	135,55	135,17	131,10	110,54
Concert	Avg	0,71	0,70 (2)	0,63	0,56 (2∗)	0,68	0,65	0,66	0,63 (2∗)	0,69	0,66	0,69	0,66 (2)
	Max	6,82	6,09	7,00	3,28	6,65	4,04	5,85	3,59	6,54	4,45	6,43	4,60
Titles	Avg	117,94	115,77	72,13	71,60	93,15	92,46	70,48	69,62	92,54	91,85	108,09	103,09
	Max	231,10	229,15	145,09	139,69	181,30	179,56	139,82	136,54	183,50	179,03	213,42	207,80
Volcano	Avg	1,16	1,16	1,18	1,17 (2)	1,10	1,10	1,02	1,02	1,14	1,14	1,12	1,12
	Max	1,88	1,88	1,90	1,88	1,76	1,76	1,65	1,64	1,85	1,85	1,82	1,82
Anemone	Avg	27,99	27,98	27,84	27,88 (2)	27,93	27,92	27,79	27,79	27,95	27,94	27,91	27,91
	Max	78,29	78,28	78,24	78,28	78,16	78,15	78,01	78,04	78,22	78,20	78,23	78,28
BBBunny	Avg	72,56	72,39 (2)	72,39	72,34 (2)	71,25	70,90 (2)	70,93	70,81	72,01	71,86	72,37	72,02 (2)
	Max	288,17	287,28	287,61	287,04	283,84	282,65	283,71	282,06	286,77	285,28	287,92	285,76
Theatre	Avg	32,75	32,71 (2∗)	32,92	32,91 (3∗)	30,84	30,89	28,04	28,09	31,90	31,91	31,35	31,36
	Max	47,54	47,58	48,05	47,76	45,79	45,57	41,81	41,81	46,51	46,32	45,34	45,49
Diver	Avg	4,45	4,44 (2)	4,46	4,45 (2∗)	4,27	4,26	4,32	4,31	4,33	4,32	4,42	4,41 (2)
	Max	12,30	12,25	12,34	12,35	11,85	11,83	11,88	11,88	12,07	12,07	12,20	12,17

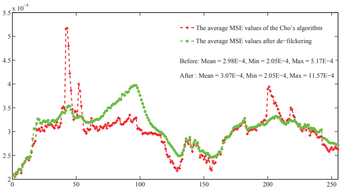


Fig. 8. MSE of the *Stars2* sequences before and after de-flickering.

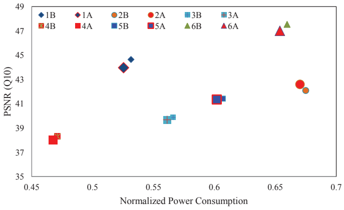


Fig. 9. The average PSNR vs. power consumption for different sequences before and after filtering. The numbers are: 1: GD with $\epsilon = 0.0002$ [2], 2: Albrecht [6], 3: Cho [5], 4: Nadernejad [1], 5: Zhang [4], and 6: GD with $\epsilon = 0.001$ [2], respectively. A and B show before and after filtering results.

References

[1] E. Nadernejad, N. Burini, J. Korhonen, S. Forchhammer, and C. Mantel, "Adaptive local backlight dimming algorithm based on local histogram and image characteristics," *Proceedings of SPIE, the International Society for Optical Engineering*, vol. 8652, p. 86520V, 2013.

[2] N. Burini, E. Nadernejad, J. Korhonen, S. Forchhammer, and X. Wu, "Speedup of optimization-based approach to local backlight dimming of hdr displays," *Proceedings of SPIE, the International Society for Optical Engineering*, vol. 8436, p. 84360B, 2012.

[3] ——, *Image Dependent Energy-Constrained Local Backlight Dimming*, ser. International Conference on Image Processing. Proceedings. IEEE, 2012, pp. 2797–2800.

[4] 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, 2012.

[5] H. Cho, B. C. Cho, H. J. 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 and Laser Technology*, vol. 47, pp. 80–87, 2013.

[6] 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 Trans. Electron. (Japan)*, vol. E93-C, no. 11, pp. 1556–1563, 2010.

[7] X. Shu, X. Wu, and S. Forchhammer, "Optimal local dimming for lc image formation with controllable backlighting," *I E E E Transactions on Image Processing*, vol. 22, no. 1, pp. 166–173, 2013.

[8] J. H. Lee, S.-E. Kim, T.-H. Lee, W.-J. Song, M. K. Kim, T. W. Lee, and C. G. Kim, "P-54: Flicker reducing backlight control based on adaptive moving average filtering," *SID Symposium Digest of Technical Papers*, vol. 40, no. 1, pp. 1294–1297, 2009. [Online]. Available: http://dx.doi.org/10.1889/1.3256533

[9] H. Chen, J. Sung, T. Ha, and Y. Park, "Locally pixel-compensated backlight dimming on led-backlit lcd tv," *Journal of the Society for Information Display*, vol. 15, no. 12, pp. 981–988, 2007. [Online]. Available: http://dx.doi.org/10.1889/1.2825108

[10] "Sita sings the blues." [Online]. Available: www.sitasingstheblues.com.

[11] N. Burini, E. Nadernejad, J. Korhonen, S. Forchhammer, and X. Wu, "Modeling power-constrained optimal backlight dimming for color displays," *Journal of Display Technology*, 2013. [Online]. Available: http://dx.doi.org/10.1109/JDT.2013.2253544

[12] "Consumer digital video database." [Online]. Available: http://www.cdvl.org.

[13] X. Fan, W. Gao, Y. Lu, and D. Zhao, "Jvt-e070 flicking reduction in all intra frame coding," JVT, Tech. Rep., 2002.

A.6 Wavelet-based Image Enhancement using Fourth Order PDE

E. Nadernejad and S. Forchhammer, “Wavelet-based Image Enhancement Using Fourth Order PDE”, in *2011 IEEE 7th International Symposium on Intelligent Signal Processing (WISP)*, IEEE, 2011, pp. 116–121

Reference: [6]

Wavelet-based Image Enhancement Using Fourth Order PDE

Ehsan Nadernejad
Department of Photonics Engineering,
Technical University of Denmark,
Kgs. Lyngby, Denmark
ehsa@fotonik.dtu.dk

Søren Forchhammer
Department of Photonics Engineering,
Technical University of Denmark,
Kgs. Lyngby, Denmark
sofo@fotonik.dtu.dk

Abstract—**The presence of noise interference signal may cause problems in signal and image analysis; hence signal and image de-noising is often used as a preprocessing stage in many signal processing applications. In this paper, a new method is presented for image de-noising based on fourth order partial differential equations (PDEs) and wavelet transform. In the existing wavelet thresholding methods, the final noise reduced image has limited improvement. It is due to keeping the approximate coefficients of the image unchanged. These coefficients have the main information of the image. Since noise affects both the approximate and detail coefficients, in this research, the anisotropic diffusion technique for noise reduction is applied on the approximation band to alleviate the deficiency of the existing wavelet thresholding methods. The proposed method was applied on several standard noisy images and the results indicate superiority of the proposed method over the existing wavelet-based image de-noising, anisotropic diffusion, and wiener filtering techniques.**

Keywords; Image Enhancement, PDE, Wavelet

I. Introduction

Noise reduction is usually the first process that is used in the analysis of digital images. In any image de-noising algorithm, it is very important that the de-noising process has no blurring effect on the image, and makes no changes or relocation to image edges.

There are various methods for image de-noising. Simple mathematical filters such as average filter, median filter and Gaussian filter are some of the techniques employed for image de-noising. Also, wavelet thresholding is a method which has been reported in several studies for noise reduction [1, 2]. In this approach, a threshold is applied on wavelet coefficients. As an advantage of this method, the edges of the image will be preserved; however, the texture of the image can be destroyed at the same time owing to the transformation of pixel values of a region with little variations to a constant value. Since images are considered as non-stationary processes [3], adaptive filters are mostly applicable in image noise reduction. Some of these filters can be designed using partial differential equations ($PDEs$) which are employed as an efficient method for noise reduction in many studies.

These equations are applied to solve the problem with the edge smoothing effect that exists in mean and Gaussian filters [4]. Two equations have been introduced in [5] for non-linear diffusion process. These equations apply low-pass filters on the image which smooth the regions with low gradient and preserve high gradient regions such as edges. The existing diffusion-based methods suggest the use of side neighbors [6]. In different studies it is shown that these methods may also lead to smoothing the edges [7, 8], however, this effect is less than the one obtained by mean and Gaussian filters [9].

In this paper, both wavelet thresholding and PDE methods are applied for image de-noising. In the proposed approach, the image is initially decomposed using wavelet transform. Then, the sub-bands containing the detail coefficients are thresholded using wavelet thresholding method, and the anisotropic diffusion is applied on approximate coefficients.

The rest of the paper is organized as follows: the concept of fourth order partial differential equations is introduced in Section 2. Section 3 describes the wavelet thresholding method. The proposed image de-noising method is presented in Section 4. Section 5 introduces various metrics used for performance measuring algorithms in this paper. The experimental results are provided in Section 6. Finally, conclusions are drawn in Section 7.

II. Fourth Order Partial Diffrential Equaiotns

In the past few years, a number of authors have proposed fourth order $PDEs$ for edge detection and image denoising with the hope that these methods would perform better than their second order analogues [10, 11]. Indeed there are good reasons to consider fourth order equations. Firstly, fourth order linear diffusion dampens oscillations at high frequencies much faster than second order diffusion. Secondly, there is the possibility of having schemes that include the effects of curvature in the dynamics, thus creating a richer set of functional behaviors. On the other hand, the theory of fourth order nonlinear $PDEs$ is far less developed than their second order analogues. Also such equations do not possess a maximum principle or comparison principle, and the implementation of the equations could thus introduce artificial singularities or other undesirable behavior [12].

In this paper, we implemented and tested the fourth order $PDEs$ proposed in [13] and [14]. For these methods, the fourth order $PDEs$, uses the L^2 – curvature gradient flow method as below:

$$\frac{\partial I}{\partial t} = -\nabla^2[c(\nabla^2 I)\nabla^2 I] \qquad (1)$$

Where ∇^2 is the *Laplacian* of the image$I$. Since the *Laplacian* of an image at a pixel is zero if the image is planar in its

978-1-4577-1402-3/11/$26.00 ©2011 IEEE

neighborhood, these $PDEs$ attempt to remove noise and preserve edges by approximating an observed image with a piecewise planar image. The equation (1) was associated with the following energy functional:

$$E(I) = \int_{\Omega} f(|\nabla^2 I|)\partial x \partial y \qquad (2)$$

Where Ω is the image support and ∇^2 denotes the Laplacian operator. Since $f(|\nabla^2|)$ is an increasing function of at $|\nabla^2|$, its global minimum at $|\nabla^2| = 0$. Consequently, the global minimum of $E(I)$ occurs when:

$$|\nabla^2 I| = 0 \quad for \quad all(x, y) \in \Omega \qquad (3)$$

A planar image obviously satisfies (1) [14], hence it provides a global minimum of $E(u)$. Planar images are the only global minimum of $E(u)$ if:

$$f^{"}(s) \geq 0 \quad for \quad all \quad s \geq 0 \qquad (4)$$

because the cost function $E(u)$ is convex under this condition. Therefore, the evolution of (1) is a process in which the image is increasingly smoothed until it becomes a planar image. But in the case of second order anisotropic diffusion, $f"(s)$ may not be greater than zero, and as a result the image evolves towards a step image and that is why it suffers from blocky effects. So images processed by fourth order $PDEs$ will look less blocky than that processed by second order anisotropic diffusion.

In [13] two different functions have been proposed to measure the oscillations in the noisy data:

$$E_1(I) = \int_{\Omega} (|I_{xx}| + |I_{yy}|)\partial x \partial y \qquad (5)$$

or

$$E_2(I) = \int_{\Omega} \sqrt{|I_{xx}|^2 + |I_{xy}|^2 + |I_{yx}|^2 + |I_{yy}|^2}\,\partial x \partial y \qquad (6)$$

The main difference between the two functions is that $E_2\ (u)$ is rotational invariant while $E_1(u)$ is not. However the implementation with $E_1(u)$ is more simple for higher dimensional problems. Based on the above functions *Lysaker* proposed the following *4th* order PDE:

$$\frac{\partial I}{\partial t} = -\left(\frac{I_{xx}}{|I_{xx}|}\right)_{xx} - \left(\frac{I_{yy}}{|I_{yy}|}\right)_{yy} - \alpha(I - I_0) \qquad (7)$$

where:

$$\alpha = -\frac{1}{\sigma^2}\int_{\Omega}\left(\frac{I_{xx}}{|I_{xx}|}(I - I_0)_{xx} + \frac{I_{yy}}{|I_{yy}|}(I - I_0)_{yy}\right)\partial x \partial y \qquad (8)$$

This method offers similar results when compared with the method proposed in [14].

III. Wavelet Thresholding Method

In recent years, many researchers have performed wavelet thresholding for image de-noising [1-3, 17-19]. Thresholding is a simple non-linear technique operating on the wavelet coefficients. In this technique, each coefficient is thresholded by comparing it to a threshold value. If the coefficient is smaller than the threshold, it is set to zero; otherwise it is kept or modified as defined in the used thresholding method. The motivation is that as the wavelet transform is good at energy compaction, the small coefficients represent noise and large coefficients indicate the important signal features [18, 19]. The small coefficients can be thresholded without affecting the significant features of the image. By applying inverse wavelet transform on the thresholded coefficients, a reconstructed image with lower noise is obtained.

In many applications, a soft thresholding method is more desirable than the hard thresholding, because it gives visually more pleasant images. The hard thresholding may lead to discontinuities and abrupt artifacts in the de-noised images, especially when the noise energy is significant. The brief description of the hard thresholding is as follows:

$$\gamma(Y) = \begin{cases} Y & if \ \ Y > T \\ 0 & otherwise \end{cases} \qquad (9)$$

where $\gamma(Y)$ is the soft thresholding function applied to coefficient Y, and T indicates the threshold value. Using this function, all coefficient values smaller than the threshold in each sub-band are replaced by zeros, and the values greater than the threshold are unchanged.

In soft thresholding, values less than the threshold are replaced by zeros and values above the threshold are reduced by the level of the thresholding using the following equation: (See Fig.1):

$$\eta(Y) = sign(Y)\max((|Y| - T), 0) \qquad (10)$$

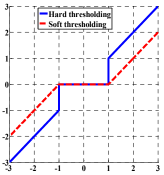


Fig.1 Thresholding functions for the noise cancelling operations (T=1)

Three methods are presented to calculate the threshold value, namely *Visushrink*, *Bayesshrink* and *Sureshrink* [1]. *Visushrink* is based on applying the universal threshold proposed by *Donoho* and *Johnstone* [2]. The threshold is given by $\sigma\sqrt{2logM}$, where σ is the standard deviation of the noise and M is the number of pixels in the image. This threshold does not adapt well with discontinuities in the image. *SureShrink* is also a practical wavelet procedure, but it uses a local threshold estimated adaptively for each level [19]. The *Bayesshrink* rule uses a *Bayesian* mathematical framework for images to derive subband-dependent thresholds. These thresholds are nearly optimal for soft thresholding, because the wavelet coefficients in each subband of a natural image can be approximated adequately by a Generalized Gaussian Distribution (*GGD*) [2].

IV. Proposed Algorithm

As it was mentioned before, in wavelet thresholding method, the detail coefficients are thresholded after decomposing the original image. In detail sub-bands, the coefficients which are less than the threshold are eliminated. This method is fast but is not efficient for image de-noising in some cases such as in images corrupted by salt and pepper or speckle noise [1, 2, 18, 19]. The large pixel values corrupted by noise cannot be removed by thresholding. By increasing the threshold value, the main structures embedded in the original image may be destroyed. On the other hand, *PDEs* are able to remove the noises such as speckles [18]. In order to increase the performance of the existing method, both wavelet thresholding and fourth-order diffusion techniques are considered in the proposed algorithm.

The numerical solution of the proposed algorithm is described below:

Let the time step be Δt and the spatial step be h in x, y directions, then the time and space coordinates can be described as:

$$\begin{aligned} & t = n\Delta, n = 0,1,2,....;\ \ x = ih, y = jh, \\ & i = 1,2,3,...M-1, j = 0,1,2,...N-1 \end{aligned} \tag{11}$$

where $Mh \times Nh$ is the size of the image. Let $I^n_{i,j} = I(ih, jh, n\Delta t)$ then the final image can be obtained using the eight-stage approach described below:

***Stage I*:** 1- Mother wavelet selection and then, computing the wavelet decomposition of the image I at level L.

2- For each level from I to L, select a suitable threshold based on *Bayesshrink* rule and apply soft thresholding to the detail coefficients.

Stage II: Let $A(x, y, t)$ be the approximate coefficients of the image with coordinates (x, y).The derivative approximations is computed from the approximate coefficient of the image:

$$\begin{aligned} & \nabla_N A^n_{i,j} = (A^n_{(i-1,j)} - A^n_{(i,j)})/h,\ \nabla_S A^n_{i,j} = (A^n_{(i+1,j)} - A^n_{(i,j)})/h \\ & \nabla_W A^n_{i,j} = (A^n_{(i,j-1)} - A^n_{(i,j)}/h,\ \ \nabla_E A^n_{i,j} = (A^n_{(i,j+1)} - A^n_{(i,j)}/h \end{aligned} \tag{12}$$

In order to solve partial differential equation, boundary condition should be used. In this paper, the Neumann boundary conditions are used [21]. These boundary conditions specify the values the derivative of the solution would take on the boundary of the domain. The symmetric boundary conditions are used:

$$\begin{aligned} & A^n_{-1,j} = A^n_{0,j}, A^n_{M,j} = A^n_{M-1,j}, j = 0,1,2,....N-1, \\ & A^n_{i,-1} = A^n_{i,0}, A^n_{i,N} = A^n_{i,N-1}, i = 0,1,2,....M-1, \end{aligned} \tag{13}$$

Stage III: Computing the diffusion coefficient $c(x, y; t)$[7] for example:

$$c^n_{i-1,j} = \exp(-[\frac{\nabla_N A^n_{i,j}}{k}]^2) \tag{14}$$

The other diffusion coefficients can be obtained similarly.

Stage IV: Computing the divergence of $c(.)\nabla A$:

$$\begin{aligned} d^n_{i,j} = & \frac{1}{h^2}[c^n_{i+1,j}(A^n_{i+1,y} - A^n_{x,y}) + c^n_{i-1,j}(A^n_{i-1,y} - A^n_{x,y}) \\ & + c^n_{i,j+1}(A^n_{i,y+1} - A^n_{x,y}) + c^n_{i,j-1}(A^n_{i,y-1} - A^n_{x,y})] \end{aligned} \tag{15}$$

With symmetric boundary conditions:

$$\begin{aligned} & d^n_{-1,j} = d^n_{0,j}, d^n_{M,j} = d^n_{M-1,j}, j = 0,1,2,....N-1, \\ & d^n_{i,-1} = d^n_{i,0}, d^n_{i,N} = d^n_{i,N-1}, i = 0,1,2,....M-1, \end{aligned} \tag{16}$$

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

$$A^{n+1}_{i,j} = A^n_{i,j} + \frac{\Delta t}{4} d^n_{i,j} \tag{17}$$

***Stage VI*:** Calculate *Laplacian* and *derivative* from the $X = A^{n+1}_{i,j}$:

$$\nabla^2 X^n_{i,j} = (X^n_{(i+1,j)} + X^n_{(i-1,j)} + X^n_{(i,j+1)} + X^n_{(i,j-1)} - 4 \times X^n_{(i,j)})/h^2. \tag{18}$$

***Stage VII*:** The final stage is calculated as below:

$$X^{n+1}_{i,j} = A^n_{i,j} - \frac{\Delta t}{4} \nabla^2 X^n_{i,j} \tag{19}$$

This equation is equivalent to (1) (in this paper: $\Delta t / 4 \approx 0.125$).

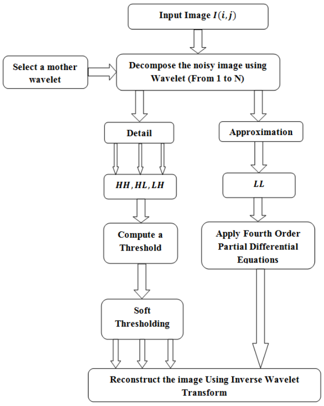

Input Image $I(i,j)$
Select a mother wavelet
Decompose the noisy image using Wavelet (From 1 to N)
Detail
Approximation
HH, HL, LH
LL
Compute a Threshold
Apply Fourth Order Partial Differential Equations
Soft Thresholding
Reconstruct the image Using Inverse Wavelet Transform

Fig.2 Structure of the proposed method

It has to be noted that in any *PDE*-based method the number of algorithm iterations is crucial. In other words, extra iteration may cause the pixel value to exceed the grey level value, relocating the edges and blurring the image; on the contrary, few iterations lead to the image that cannot be de-noised efficiently.

Final Stage: Applying the inverse wavelet transform on sub-bands to obtain the de-noised image.

V. Performance Meseares

In this paper three well known metrics are used to evaluate algorithms, which are introduced here briefly.

A: Figure of Merit Metric

The figure of merit (*FOM*) edge preserving measure is defined as [20]:

$$FOM = \frac{1}{\max\left\{\hat{N}, N_{ideal}\right\}} \sum_{i=1}^{\hat{N}} \frac{1}{1 + d_i^2 \lambda} \tag{20}$$

In this equation $\hat{N}$ and N_{ideal} are the numbers of detected and original edge pixels, respectively; d_i is the Euclidean distance between the i th detected edge pixel and the nearest original edge pixel; λ is a constant typically set to 1/9. The dynamic range of *FOM* is between the processed image and the ideal image. We used the *Canny* edge detector [21] to find the edge in all processed results.

B: Structural SIMilarity Metric (*MSSIM*)

The Structural Similarity (*SSIM*) factor can be used to measure the similarity of two images [22]. This factor consists of three different metrics. Let $X = \{x_i | i = 1,2,3, \dots, N\}, Y = \{y_j | j = 1,2,3, \dots, N\}$ be the original and the noisy image, respectively. Then, *SSIM* is defined as:

$$SSIM = \left(\frac{\sigma_{xy}}{\sigma_x \sigma_y}\right)^{\alpha} \times \left(\frac{2\overline{X}\overline{Y}}{(\overline{X})^2 + (\overline{Y})^2}\right)^{\beta} \times \left(\frac{2\sigma_x \sigma_y}{\sigma_x^2 + \sigma_y^2}\right)^{\gamma} \\ = s(x,y)^{\alpha} \times l(x,y)^{\beta} \times c(x,y)^{\gamma} \tag{21}$$

where $\alpha > 0, \beta > 0$ and $\gamma > 0$ are parameters. The *SSIM* indicates any distortion as a combination of three different factors: loss of correlation, luminance distortion, and contrast distortion. The first component is the correlation coefficient between *x* and *y*, representing the degree of linear correlation between *x* and *y*, and its dynamic range is between -1 and 1. The highest possible value 1 is obtained when $y_i = ax_i + b$ for all $i = 1,2,\dots,N$, where a and b are constants and $a > 0$. The second component, with a range of [0, 1], measures how close *x* and *y* are in luminance. It equals 1 if and only if $\overline{X} = \overline{Y}$. σ_x and σ_y can be considered as an estimate of the contrast in x and y. Eventually, the third component indicates how similar the contrast levels of the images are. Its value ranges between 0 and 1, where the highest value 1 is achieved if and only if $\sigma_X = \sigma_Y$.

The parameters are used to adjust the relative importance of the three conditions above. In order to simplify the expression, we set $\alpha = \beta = \gamma = 1$ in this paper.

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 $i = 1,2,3,\dots,N$. The lowest value of -1 occurs when $y_i = 2\bar{X} - x_i$ for all $i = 1,2,3,\dots,N$. For $\alpha = \beta = \gamma = 1$, *SSIM* in (21) can be rewritten as:

$$SSIM = \frac{4\sigma_{xy}\overline{X}\overline{Y}}{(\sigma_x^2 + \sigma_y^2)[(\overline{X})^2 + (\overline{Y})^2]} \tag{22}$$

In practice, to use the measure described above, the image is windowed equally, and *SSIM* is then computed for each window to find the average *SSIM* as follows:

$$MSSIM = \overline{SSIM}(X,Y) = \frac{1}{M}\sum_{j=1}^{M} SSIM(x_j, y_j) \tag{23}$$

where *X* and *Y* are the original and the de-noised images respectively, *M* is the number of local windows in the image, x_j and y_j are the image contents at the j^{th} local window.

C. Peak signal to noise ratio

The *PSNR* is a performance metric indicating the ratio between the maximum possible power of a signal and the power of the corrupting noise that affects the fidelity of its representation. It is most commonly used as a measure of the quality of a de-noised signal or image. It is most easily defined via the mean squared error (*MSE*) that is defined for two $v \times u$ images X and Y as follows:

$$MSE = \frac{1}{vu}\sum_{i=0}^{v-1}\sum_{j=0}^{u-1}\left|X(i,j) - Y(i,j)\right| \tag{24}$$

The *PSNR* is defined as:

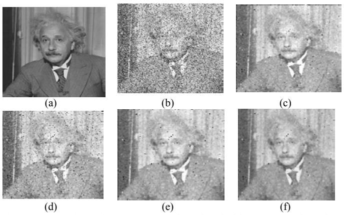
$$PSNR = 10\log_{10}\left(\frac{(MAX(X))^2}{MSE}\right) = 20\log_{10}\left(\frac{MAX(X)}{\sqrt{MSE}}\right) \tag{25}$$

where *MAX*(*X*)=255 is the maximum possible value of a pixel.

VI. Simulation Results

To compare the performance of the above-described techniques in image de-noising, they have been implemented using *Matlab*. Then the algorithms were applied to 100 standard images provided in [23].

The evaluation results obtained from one of the images are shown in Fig. 3. In the evaluation process, the image in Fig. 3a was initially affected by a combination of Gaussian noise (m=0.2, σ=0.1) and salt and pepper (density=0.15) noises, and the noisy image is shown in Fig. 3b. This image was initially decomposed using a five-level wavelet transform. The decomposition was performed using Daubechies10 mother wavelet. This mother wavelet has a good performance in noise reduction as reported in [24]. The soft thresholding method was applied on the detail coefficients in each level. The fourth order diffusion equation was applied on the approximate coefficients as described earlier. Finally, by applying the inverse wavelet transform, the de-noised image was obtained.



(a) (b) (c)

(d) (e) (f)

Fig.3 Comparison between wavelet thresholding, second order diffusion, fourth-order PDE and the proposed method in reducing Gaussian added to salt & pepper noise in one image, (a) Original image, (b) Noisy image (Gaussian and salt & pepper noise), (c) Second order diffusion, (d) Wavelet thresholding, (e) Fourth-order PDE method, (f) Proposed method.

The results of enhancing the Fig. 3b by using the five approaches are shown in Figs. 3c–f. To numerically compare the performance of the proposed method with the existing wavelet thresholding approach, the quality index and *PSNR* criteria are used. The *SSIM* and *PSNR* for the de-noised image using the proposed method shown in Fig. 4f are, respectively, 0.776 and 25.2 and for the second order diffusion shown in Fig. 4c they are 0.65 and 22.4, respectively.

The corresponding values for the wavelet thresholding method shown in Fig. 3d are 0.442 and 16.3 and the fourth-order *PDE* in Fig. 3e are 0.478 and 17.1.
To further evaluate the performance of the proposed method, the five considered methods were applied on a number of standard images from [23].

At the beginning all the images have been converted into grey scale images. We had two images, 256×256 and 512×512, for each picture. In the evaluation process, the images were corrupted by Gaussian noise ($m = 0.2, \sigma = 0.15$). The quality index and *PSNR* criteria for the enhanced images using the four approaches, wavelet thresholding, second order diffusion, fourth-order diffusion and the proposed method, were depicted in Fig. 4. It is shown that the proposed method shows a better performance in enhancing noisy images.

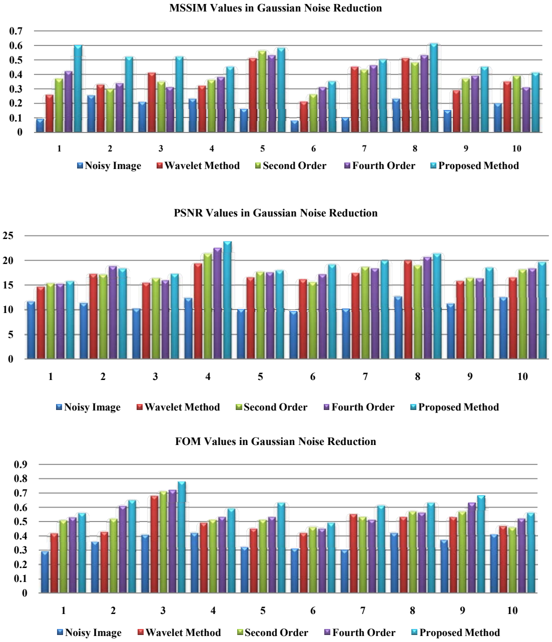


Fig.4 Quality index (*MSSIM*), *PSNR* and *FOM* criteria for comparing performance of the proposed method with the existing approach in reducing noise from different images (Gaussian Noise m=0.2,σ=0.1).

VII. Conclusion

Wavelet thresholding is used in many image de-noising techniques. In these approaches the thresholding is not applied on the approximate sub-bands, which have the main information about the signals. Since noise affects both the approximate and detail sub-bands of images, the de-noised images using these approaches have restriction in improvement. In this study, a new approach using the combination of wavelet thresholding and fourth order partial differential equations was introduced for image de-noising. In the proposed method, both thresholding and fourth order diffusion methods are combined to reduce the noise in all sub-bands of the image. Results obtained on a number of standard images illustrate that the proposed method presents a better performance compared to the existing wavelet thresholding method in image de-noising.

References

[1] S.G.Chang, B.Yu, M. Vetterli, “Adaptive wavelet thresholding for image denoising and compression,” *IEEE Trans. Image Process.*, 2000, 9, (9), pp. 1532–1545.

[2] D.L Donoho, I.M. Jonstone,”Ideal spatial adaptation via wavelet shrinkage,” *Biometrika*, 1994, 81, pp. 425–455.

[3] N. Azzabou, N. Paragios, F. Guichard,”Uniform and textured regions separation in natural images towards MPM adaptive denoising,” *Int. Conf. on Scale Space and Variational Methods in Computer Vision*, 2007,pp. 418–429.

[4] E. Nadernejad, H. Hassanpour, H. Miar,”Image restoration using a PDE-based approach,” *Int. J. Engng., Trans. B*:Appl., 2007, 20, (3), pp. 225–236.

[5] P.Perona, J. Malik,”Scale-space and edge detection using anisotropic diffusion,”*IEEE Computer Society Workshop on Computer Vision,* 1987, pp. 16–27.

[6] G.Gilboa, N. Sochen, Y.Y. Zeevi, “Image enhancement and denoising by complex diffusion process,” *IEEE Trans. Patt. Anal*. Mach. Intell., 2004, 26, (8), pp. 1020–1036.

[7] H.Hassanpour, E. Nadernejad, H.Miar,”Image enhancement using diffusion equations.”. *Int. Symp. On Signal Processing and its Applications (ISSPA)*, Sharjah, UAE, 2007.

[8] Nadernejad. E, Hassanpour. H,”A comparison and analysis of different PDE-based approaches for image enhancement,” *Int. Conf. on Signal Processing and Communication Systems (ICSPCS),* Australia, 2007.

[9] Luo H., Zhu L., Ding H.: ‘Coupled anisotropic diffusion for image selective smoothing’, *Elsevier Signal Process*., 2006, 86, pp. 1728–1736.

[10] Yu-Li You and M. Kaveh, “Fourth order partial differential equations for noise removal,” *IEEE Trans. Image Processing,* vol. 9, no. 10, 2000, pp 1723-1730.

[11] T. Chan, A. Marquina, and P. Mulet. “High-order total variation based image restoration”. *SIAM J. Sci. Comp., 22(2), 2000, pp 503–516.*

[12] J.B. Greer and A. L. Bertozzi, “Travelling wave solutions of fourth order $PDEs$ for image processing”, *SIAM Journal on Mathematical Analysis,* vol. 36, no. 1, 2004.

[13] Y.L. You, X. Wenguan, A. Tannenbaum, M. Kaveh, “Behavioral analysis of anisotropic diffusion in image processing,” *IEEE Trans. Image Processing,* vol. 5, no. 11, 1996, pp 1539-1553.

[14] M. Lysaker, A. Lundervold, X.C. Tai, “Noise removal using fourth order partial differential equation with applications to medical magnetic resonance images in space and time,” *IEEE Trans. Image Processing,* vol. 12, no. 12, 2003, pp 1579 – 1590.

[15] A.F.Koschan, M.A. Adibi,”A comparison of median filter techniques for noise removal in color images,” *Proc. Seventh German Workshop on Color Image Processing*, Erlangen, Germany, October 2001, pp. 69–79

[16] G.Aubert, P.Kornprobst,”Mathematical problems in image processing: partial differential equations and calculus of variations,”*Springer*, New York, 2005.

[17] C.S.Burrus, R.A.Gopinath, H.Guo, “Introduction to wavelets and wavelet transforms,” (Prentice-Hall, New Jersey, 1998).

[18] S.G.Chang, B.Yu, M.Vettereli,”Adaptive wavelet thresholding for image denoising and compression,” *IEEE Trans. Image Process.*, 2000, 9, (9), pp. 1532–1546.

[19] D.L.Donoho, I.M. Johnstone, G.Kerkyacharian, D.Picard, “Wavelet Shrinkage: Asymptopia?,”*, J. R oyal Stat. Soc*., Ser.B, 1995, 57, (2), pp. 301–369.

[20] W.K.Pratt, "Digital Image Processing", *Wiley, New York, NY, USA,* 2001.

[21] Yu Y, Acton S. “Speckle reducing anisotropic diffusion*”. IEEE Trans Image Process* 2002;11(11):1260–70.

[22] Z.Wang and A.C.Bovik, "A universal image quality index," *IEEE Signal Processing letters, vo.9,no.3,pp-81-84, 2002.*

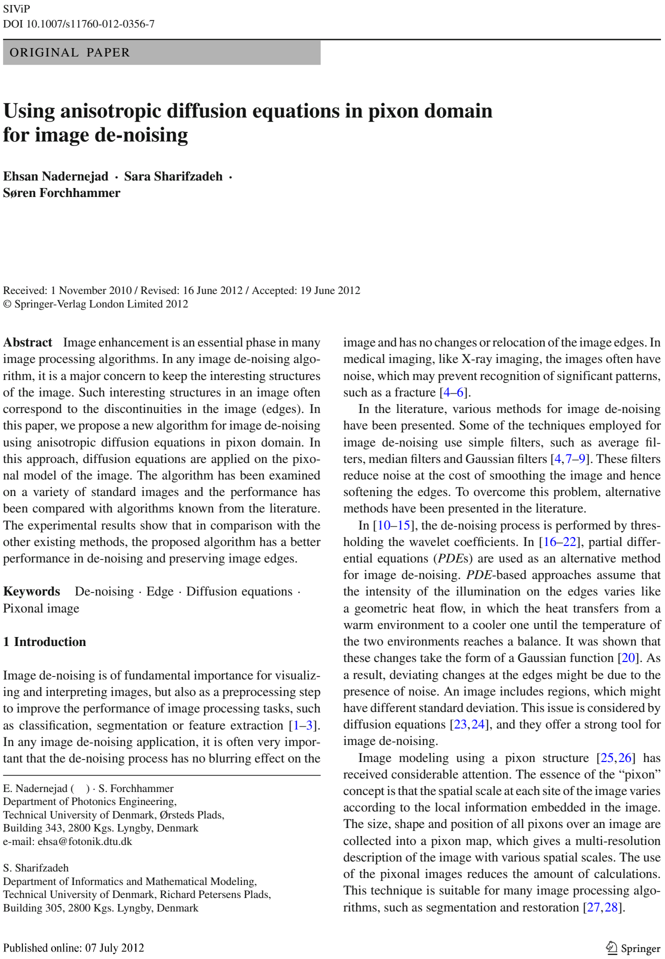
[23] R.C. Gonzalez, R. E.Woods, “Image Databases,“ at available : <http://www.imageprocessingplace.com>.

[24] J.H.Zhai, S.F.Zhang, “Image denoising via wavelet threshold: single wavelet and multiple wavelets transform,’ *Proc. Fourth Int. Conf. on Machine Learning and Cybernetics*, Guangzhou, 2005.

A.7 Using Anisotropic Diffusion Equations in Pixon Domain for Image Denoising

E. Nadernejad, S. Forchhammer, and S. Sharifzadeh, “Using Anisotropic Diffusion Equations in Pixon Domain for Image De-noising”, *Signal, Image and Video Processing*, 2012, ISSN: 1863-1703

Reference: [7]



SIViP
DOI 10.1007/s11760-012-0356-7

ORIGINAL PAPER

Using anisotropic diffusion equations in pixon domain for image de-noising

Ehsan Nadernejad · Sara Sharifzadeh · Søren Forchhammer

Received: 1 November 2010 / Revised: 16 June 2012 / Accepted: 19 June 2012

Abstract Image enhancement is an essential phase in many image processing algorithms. In any image de-noising algorithm, it is a major concern to keep the interesting structures of the image. Such interesting structures in an image often correspond to the discontinuities in the image (edges). In this paper, we propose a new algorithm for image de-noising using anisotropic diffusion equations in pixon domain. In this approach, diffusion equations are applied on the pixonal model of the image. The algorithm has been examined on a variety of standard images and the performance has been compared with algorithms known from the literature. The experimental results show that in comparison with the other existing methods, the proposed algorithm has a better performance in de-noising and preserving image edges.

Keywords De-noising · Edge · Diffusion equations · Pixonal image

1 Introduction

Image de-noising is of fundamental importance for visualizing and interpreting images, but also as a preprocessing step to improve the performance of image processing tasks, such as classification, segmentation or feature extraction [1–3]. In any image de-noising application, it is often very important that the de-noising process has no blurring effect on the image and has no changes or relocation of the image edges. In medical imaging, like X-ray imaging, the images often have noise, which may prevent recognition of significant patterns, such as a fracture [4–6].

In the literature, various methods for image de-noising have been presented. Some of the techniques employed for image de-noising use simple filters, such as average filters, median filters and Gaussian filters [4,7–9]. These filters reduce noise at the cost of smoothing the image and hence softening the edges. To overcome this problem, alternative methods have been presented in the literature.

In [10–15], the de-noising process is performed by thresholding the wavelet coefficients. In [16–22], partial differential equations (*PDEs*) are used as an alternative method for image de-noising. *PDE*-based approaches assume that the intensity of the illumination on the edges varies like a geometric heat flow, in which the heat transfers from a warm environment to a cooler one until the temperature of the two environments reaches a balance. It was shown that these changes take the form of a Gaussian function [20]. As a result, deviating changes at the edges might be due to the presence of noise. An image includes regions, which might have different standard deviation. This issue is considered by diffusion equations [23,24], and they offer a strong tool for image de-noising.

Image modeling using a pixon structure [25,26] has received considerable attention. The essence of the "pixon" concept is that the spatial scale at each site of the image varies according to the local information embedded in the image. The size, shape and position of all pixons over an image are collected into a pixon map, which gives a multi-resolution description of the image with various spatial scales. The use of the pixonal images reduces the amount of calculations. This technique is suitable for many image processing algorithms, such as segmentation and restoration [27,28].

E. Nadernejad ( ) · S. Forchhammer
Department of Photonics Engineering,
Technical University of Denmark, Ørsteds Plads,
Building 343, 2800 Kgs. Lyngby, Denmark
e-mail: ehsa@fotonik.dtu.dk

S. Sharifzadeh
Department of Informatics and Mathematical Modeling,
Technical University of Denmark, Richard Petersens Plads,
Building 305, 2800 Kgs. Lyngby, Denmark

Published online: 07 July 2012

In the proposed technique, the pixonal model of the image is first obtained. Then, to reduce the noise and restore the image, anisotropic diffusion is applied to the pixonal image. It is shown in this paper that in comparison with other existing methods, the proposed algorithm has a better performance in de-noising and preserving image edges.

The rest of the paper is organized as follows. Pixonal image are described in Sect. 2. Anisotropic diffusion equations in pixon domain (ADEPD) are described in Sect. 3. Section 4 introduces various metrics used for performance measurements for the algorithms presented in this paper. The experimental results are provided in Sect. 5. Finally, conclusions are drawn in Sect. 6.

2 Pixonal image

The pixon concept for astronomical image restoration and reconstruction was proposed by *Pina* and *Puetter* [25,26] and later for image segmentation and de-noising was applied by *Yang* and *Jiang* [27,28]. We shall follow the latter version.

2.1 Pixon definition

A *pixon* is a group of connected variable size pixels, which locally represents the resolution of the image. Let $Y = \{y_i\}_{i=1}^{N}$ be the set of all pixels in an image, a subset of Y is a pixon if and only if all its pixels are connected. A pixon is then, denoted by $P_i = \left\{y_{ij}\right\}_{j=1}^{n_i}$, where n_i is the number of pixels in P_i. In addition, a set of pixons in an image forms a pixonal image if and only if each pixel belongs to exactly one pixon and no pixon is the null set.

The shape and position of all the pixons of an image are collected in a pixon map, which gives a multi-resolution description of the image with various spatial scales. Since different parts of an image often exhibit different spatial resolution, the use of a pixon map, as an adaptive scale representation, is justified [28].

In order to construct a pixonal image, we calculate a base image from which the pixons are extracted. This image is built using a kernel function and a pseudo image (I_{pseudo}).

A pseudo image is a high-resolution image, obtained by interpolation of the pixels in the $(M \times N)$ input image. A kernel function is applied to each pixel of the pseudo image to smooth the image. In this paper, a two-dimensional Gaussian kernel function has been used. Each pixel of the base image $I(x, y)$ is modeled by a local convolution of the pseudo image I_{pseudo} and the kernel function $K_{(\sigma_x,\sigma_y)}$:

$$I(x, y) = \sum_{r=-w}^{+w} \sum_{c=-w}^{+w} K_{(\sigma_x,\sigma_y)}(r, c)\, I_{\text{pseudo}}(x - r, y - c)\,, \tag{1}$$

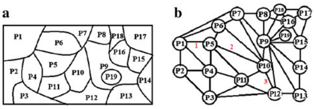


Fig. 1 **a** Pixonal image, **b** corresponding graph representation

where (x, y) is the row and column index of each pixel of the base image in the spatial domain, σ_x and σ_y are the standard deviation of the Gaussian kernel and w is the width of the Gaussian kernel. From the base image, pixons are extracted using a clustering strategy, which will be explained in the stage I in Sect. 3.2.

Following *Yang* and *Jiang* [28,29], a *pixonal image* is defined as:

$$I = \bigcup_{i=1}^{s} P_i\,, \tag{2}$$

where I is the pixonal image derived from the base image; s is the number of pixons; P_i is pixon i, composed of a set of connected pixels. This could be one pixel or a sub-pixel in the original image, as the original image has been interpolated form the pseudo image. The *pixon intensity*, $|P_i|$, is defined as the mean value of its pixels. The shape of each pixon may vary, depending on the observed image.

When the pixonal image is constructed, the image restoration problem is transferred from pixel domain to pixon domain. Therefore, s pixons are evaluated, instead of $(M \times N)$ pixels $(s \ll M \times N)$. This may reduce the computational load in image analysis.

2.2 Graphical representation

A pixonal image could be represented by a graph-like structure, in which pixons form a set of graph vertices, and an edge between two vertices indicate a common border between the two pixons. Hence, the graphical structure of a pixonal image can be represented as:

$$G = (Q, E)\,, \tag{3}$$

where Q is the finite set of vertices in the graph, representing the pixons in the image, and E is the set of the edges in the graph, indicating that the two connected pixons are neighbors. In the example in Fig. 1a, $P_1, P_2, \ldots, P_{19}$ represent nineteen pixons forming the pixonal image. The pixonal image is also illustrated by a graph structure in Fig. 1b.

2.3 Pixonal image properties

Since the lattice structure of pixels is transformed to pixons, the neighborhood structure of the image representation will change due to the fact that a pixon might consist of one pixel or tens of pixels. As a result, there might be just one neighbor for a pixon if a pixon is inside another pixon (see P_{19} in Fig. 1), or more than eight neighbors (see P_9 in Fig. 1).

In addition, using a pixonal image will reduce the effect of noise. In image de-noising, when a pixonal image is built, during the grouping of the similar pixels and calculation of the pixons value, the effect of noise is attenuated. For example, consider n pixels grouped in one pixon, where k of them are noisy ($k \ll n$). Clearly, Px is calculated out of the number of pixels where just k numbers of them are noisy. Therefore, the effect of noise in Px will be reduced [27].

3 Anisotropic diffusion in the pixonal image domain

In this paper, we propose a new image de-noising method in which anisotropic diffusion is applied on pixonal images for de-noising. Below, the anisotropic diffusion equations are described, and finally, the proposed method is presented.

3.1 Anisotropic diffusion equation

Nonlinear diffusion filtering was introduced by *Perona* and *Malik* [23,24]. Their approach in its original formulation has some deficiencies, which have encouraged many researchers to improve the technique [14,21,30–32].

The anisotropic diffusion equation belongs to the family of partial differential equations, and the solution is based on iteration. The main idea of using diffusion equations in image processing is the use of a two-dimensional Gaussian filter in multi-scale image analysis. Convolving an image with a Gaussian filter,

$$K_\sigma\,(x,y) = \frac{1}{2\pi\sigma^2}\exp\left(-\frac{|x|^2+|y|^2}{2\sigma^2}\right), \tag{4}$$

with standard deviation σ, is equivalent to the solution of the diffusion equation in two dimensions. Using the convolution operation, there would be a problem with edge smoothing. It has been shown that this problem can be resolved by treating the intensity variations in an image as diffusion of a heat flow [24].

The diffusion equation for an image $I(x,y)$ is [23]:

$$\frac{\partial I(x,y,t)}{\partial t} = \nabla^2 I(x,y,t) = \frac{\partial^2 I(x,y,t)}{\partial x^2} + \frac{\partial^2 I(x,y,t)}{\partial y^2}, \tag{5}$$

where $I(x,y,t)$ is the two-dimensional image as a function of position (x,y) at time t. We take $I\,(x,y) = I\,(x,y,t)$ at $t = 0.5\sigma^2$ as the filtered images. The initial condition is $I\,(x,y,0) = I_0(x,y)$, where I_0 is the input image. Equation (5) can be rewritten in a more general form as:

$$\begin{aligned}&\frac{\partial I\,(x,y,t)}{\partial t} = \nabla\cdot(c(x,y,t)\nabla I(x,y,t))\\ &I\,(x,y,0) = I_0\,(x,y)\,,\end{aligned} \tag{6}$$

where ∇ is the gradient operator, $c(x,y,t)$ is a diffusion function, and $\nabla.$ is the divergence operator. If c has a constant value (independent of x, y, t), the obtained equation is called the diffusion equation with isotropic diffusion factor. In this case, all the points, even the edges, would be smoothed as if there were no difference between the edge pixels and other pixels. Obviously, this is not an ideal condition. For resolving this deficiency, the diffusion function could be considered a function of x and y. Hence, the above equation is changed to a linear and anisotropic equation. If c is dependent on the image, the linear equation would be transformed into a nonlinear equation. This is the idea suggested in [23,24]. In our work, we have used the following:

$$c(x,y,t) = \frac{1}{\left(1+\frac{\|\nabla I\|^2}{k^2}\right)}. \tag{7}$$

In these equations, the diffusion function c changes at different points in the image. In the points with a large gradient, the function has a small value. Consequently, the diffusion function would be small around the edges. In (7), k is used to control the diffusion function.

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 [24]. Theoretically it is ill-posed in the sense that similar images are likely to diverge during the diffusion process [11]. In order to overcome this problem, it was suggested to use a regularized version of the image to calculate the gradient. The idea is to substitute in the diffusion coefficient the gradient of the image by a smooth version of the gradient [16,32]:

$$\nabla I_G = \nabla(G_\sigma * I\,(x,y,t)), \tag{8}$$

where G_σ is a Gaussian kernel, and $*$ is a convolution operator. In other words, the gradient is taken after the image at time t is smoothed by convolution with a Gaussian. σ can be considered as a regularization parameter which describes a uniform smoothing used to calculate the gradient of the image at time t. The main advantage of this model is that, if the initial image is very noisy (introducing large oscillations in the gradient of I), then the Perona and Malik model cannot distinguish between true edges and false edges created by the noise. The proposed model in (8) avoids this drawback, since

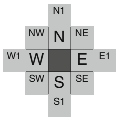


Fig. 2 The neighboring structure for the numerical solution of diffusion equations

now the equation diffuses only if the gradient is estimated to be small [21]. The Gaussian smoothing in (8) is used in our implementation (with $\sigma = 0.1$) in the first iteration.

There are a number of methods for solving diffusion equations. In [33], the first and second derivatives and also the *Laplacian* of a current pixel are computed using the neighboring pixels. The most general numerical approach to solve these equations is presented in [33]. In this paper, a similar approach is applied. To state the solution in terms of (6), Eq. (5) is modified as:

$$\frac{\partial I(x,y,t)}{\partial t} = c(x,y,t)\left(\frac{\partial^2 I(x,y,t)}{\partial x^2} + \frac{\partial^2 I(x,y,t)}{\partial y^2}\right) + \frac{\partial I(x,y,t)}{\partial x} \times \frac{\partial c(x,y,t)}{\partial x} + \frac{\partial I(x,y,t)}{\partial y} \times \frac{\partial c(x,y,t)}{\partial y}, \qquad (9)$$

where we select $c(x,y,t)$ by (7). An approximate solution to this equation can be obtained in the discrete domain. All the equations are based on a neighbor pixel structure. The proposed approach benefits from the algorithm introduced in [22,34]. It also provides a smooth change for the pixel values on the edges [21,22].

The main idea for using the neighboring structure, depicted on Fig. 2, is to preserve slant edges, in addition to vertical and horizontal edges. Using the structure (Fig. 2), the diffusion equation can, in discrete form [21,34], be written as follows:

$$\begin{aligned} I(x,y,t+\Delta t) = I(x,y,t) + \Delta t(&d_n c_n + d_s c_s + d_e c_e \\ &+ d_w c_w + \alpha \times (d_{ne} c_{ne} + d_{se} c_{se} \\ &+ d_{nw} c_{nw} + d_{sw} c_{sw}) + \alpha/2(d_{n1} c_{n1} \\ &+ d_{s1} c_{s1} + d_{e1} c_{e1} + d_{w1} c_{w1}), \end{aligned} \qquad (10)$$

where d and c are the gradient and the diffusion coefficient in each direction, respectively. The parameter α is a constant representing the significance of the slant edges in an image and Δt defines the time steps for the numerical solution. The variables c and d in (10) are given as examples:

$$d_n = I(x, y-1, t) - I(x,y,t), \quad c_n = \frac{1}{1 + \left(\frac{d_n}{k}\right)^2}. \qquad (11)$$

The other terms in (10) can be found in a similar manner. These parameters are the numerical solutions of the derivation and diffusion factors for the different directions. Having more steps and iteration levels, a smoother image will be created. The coefficients should be chosen accurately so that the obtained pixel gray levels do not exceed the standard range. In this paper, parameters $\Delta t = 0.125\text{and}\alpha = 0.5$ have been chosen as suggested in [21].

3.2 Anisotropic diffusion equation in pixon domain

In this paper, a novel pixon domain method is proposed for image de-noising. The method uses the combination of *PDE*s and the concept of pixons [25,26]. In our method, the pseudo image technique and blurring kernel function are successfully used to smooth the image and prepare it to form the pixons [27,28]. Utilizing the pseudo image and blurring function leads to the elimination of some smaller details and results in a smaller number of pixons.

In the following step, the appropriate pixons are extracted thus dividing the image into pixons using a hierarchical clustering method [35,29]. Then, image de-noising is accomplished by applying anisotropic diffusion to the pixonal image.

The numerical solution of the proposed algorithm is described as follows: Let the time step be Δt and the spatial distance h between two pixels in both x and y directions. Then, the time and space coordinates in discrete form are:

$$\begin{aligned} &t = n\Delta t, \quad n = 0, 1, 2, \ldots; \quad x = ih, y = jh, \\ &i = 1, 2, 3, \ldots, M-1, \quad j = 1, 2, 3, \ldots, N-1, \end{aligned} \qquad (12)$$

where $Mh \times Nh$ is the size of the base image. Let $I^n_{i,j} = I(ih, jh, n\Delta t)$; then, the final image can be obtained using the six-stage procedure described below:

Stage I: Obtaining the pixonal image in three steps:

I-1—Building a pseudo image with increased resolution based on bilinear interpolation.
I-2—Convolution of the pseudo image using a kernel function to obtain the base image (1). In this paper, a Gaussian blurring function is used as kernel to form the pixons [27,29].
I-3—Using a clustering algorithm to extract pixons out of the base image [29].

Details of step I: A pseudo image is made from the original image. If the original image has the dimension $M \times N$, then the dimension of the pseudo image is $lM \times lN$, where $l = 2^m$ [28,29]. m is a parameter of the algorithm.

For $m = 0$, the pseudo image is the same as the original image.

For $m > 0$ the algorithm is initialized by I_1 equal to the original image. Then at each iteration (m), a pseudo image is build out of the previous image using bilinear interpolation [29].

Parameter m is of great importance. If $m \geq 1$, the resolution of the pseudo image is larger than the original image and the final pixons may have sub-pixel accuracy. So, m determines the smallest size of a pixon. In the image parts, where the intensities of nearby pixels are similar and thus have little additional information, the intensity of newly inserted pixels have intensities similar to those of the pixels in the observed image, from which the new pixels are obtained through interpolation.

In the next step (I-2), an adaptive kernel blurring function is applied on the pseudo image and the base image is formed. The resolution of the image is increased before applying the blurring function. The following adaptive smoothing should ideally smooth the image more in homogenous regions than in texture regions. A smoothing filter as in (1), but with adaptive $\sigma\,(\sigma_x = \sigma_y)$ is applied to the pseudo image. The parameter is adaptively chosen based on local variations of the pixels. For each pixel, the adaptive standard deviation σ is calculated as the standard deviation within a 3×3 window. This step will adapt to the activity in different regions (texture, edge, etc.) and can help to avoid destroying the small scale structure of the texture area.

After forming the base image, a k-means algorithm [29] is used based on hierarchical clustering to extract pixons. For this purpose, initially each pixel represents a cluster. Then the clusters are merged according to their intensities to form larger pixons. The merging strategy is based on comparing each pixel with its 8 adjacent pixels. The mean value of the connected pixels making up the pixon is defines the pixon intensity, to form the pixonal image (See Fig. 1).

To stop the algorithm, a threshold value, T, is assigned and the merging process iterates until the difference between intensities of any two adjacent pixons are greater than the threshold value. Figure 3b shows the boundaries of the pixons as in Fig. 1a [36] (The red lines represent the boundaries of the pixons). In figures 3c, d, the pixon construction is depicted for $T = 5$ and $T = 10$ in a higher zoom. If the threshold parameter be high, the image will become smooth, which it is not suitable for de-noising purpose. In this paper, the pixonal image is build using a low threshold parameter of 5.

Stage II: at this stage, anisotropic diffusion is iteratively applied on the pixonal image for the purpose of de-noising.

Let $|P_x|$ denote the intensity of pixon P_x in the base image (See Fig. 4), the anisotropic diffusion should be applied on each pixon.

The gradient approximation for the pixonal image has been expressed as an extension of the method described in [24]. The neighbor structure in a pixonal image is not necessarily well ordered and symmetric as in pixel images shown in Fig. 1. In order to estimate the diffusion coefficients for a given pixon (P_x), first, three types of gradient elements of the pixonal image are calculated in pixon domain at iteration n:

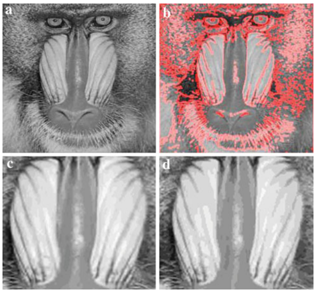


Fig. 3 **a** The original image (255 × 255), **b** corresponding boundaries between pixons, **c** and **d** two different pixon images with thresholds and 10 respectively

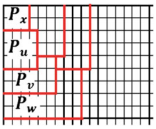
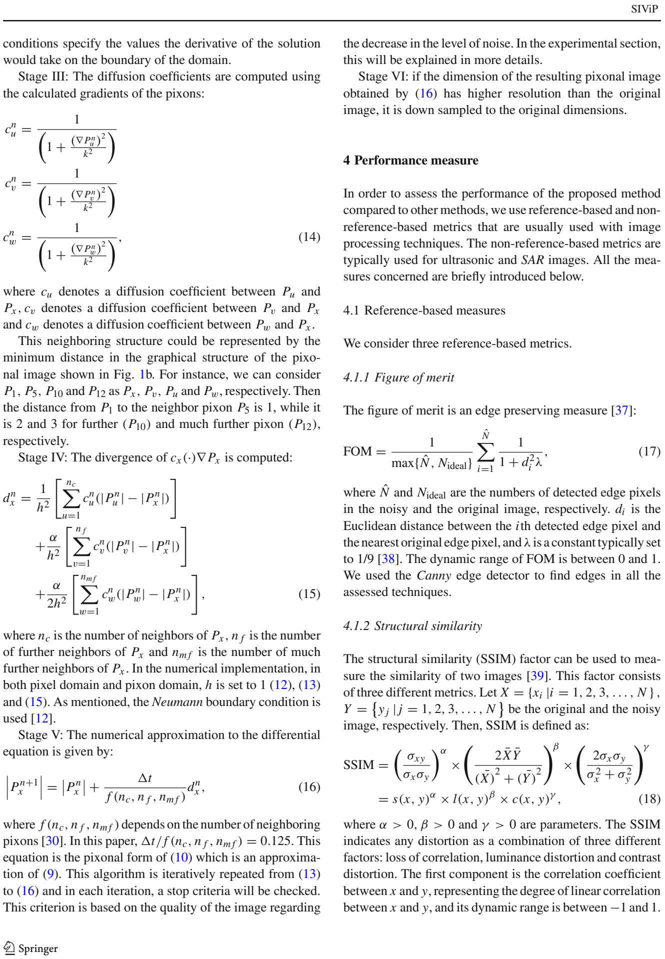


Fig. 4 Illustration of the pixonal neighboring strategy and the relation between pixons and pixels

$$\begin{aligned}\nabla P_u^n &= (\left|P_u^n\right| - \left|P_x^n\right|)/h\\ \nabla P_v^n &= (\left|P_v^n\right| - \left|P_x^n\right|)/h\\ \nabla P_w^n &= (\left|P_w^n\right| - \left|P_x^n\right|)/h,\end{aligned} \tag{13}$$

where h is the distance in pixon domain, P_u denotes a neighbor pixon of P_x which has a common boundary, P_v denotes a further neighbor of P_x which has just common boundaries with some P_u but not with P_x and P_w denotes a much further neighbor of P_x which has just common boundaries with some P_v but not with P_x or any P_u.

In order to solve the partial differential equation, boundary conditions should be used. In this paper, the Neumann boundary conditions are used [12,30]. These boundary



conditions specify the values the derivative of the solution would take on the boundary of the domain.

Stage III: The diffusion coefficients are computed using the calculated gradients of the pixons:

$$c_u^n = \frac{1}{\left(1 + \frac{(\nabla P_u^n)^2}{k^2}\right)}$$

$$c_v^n = \frac{1}{\left(1 + \frac{(\nabla P_v^n)^2}{k^2}\right)}$$

$$c_w^n = \frac{1}{\left(1 + \frac{(\nabla P_w^n)^2}{k^2}\right)}, \tag{14}$$

where c_u denotes a diffusion coefficient between P_u and P_x, c_v denotes a diffusion coefficient between P_v and P_x and c_w denotes a diffusion coefficient between P_w and P_x.

This neighboring structure could be represented by the minimum distance in the graphical structure of the pixonal image shown in Fig. 1b. For instance, we can consider P_1, P_5, P_{10} and P_{12} as P_x, P_v, P_u and P_w, respectively. Then the distance from P_1 to the neighbor pixon P_5 is 1, while it is 2 and 3 for further (P_{10}) and much further pixon (P_{12}), respectively.

Stage IV: The divergence of $c_x(\cdot)\nabla P_x$ is computed:

$$d_x^n = \frac{1}{h^2}\left[\sum_{u=1}^{n_c} c_u^n(|P_u^n| - |P_x^n|)\right] + \frac{\alpha}{h^2}\left[\sum_{v=1}^{n_f} c_v^n(|P_v^n| - |P_x^n|)\right] + \frac{\alpha}{2h^2}\left[\sum_{w=1}^{n_{mf}} c_w^n(|P_w^n| - |P_x^n|)\right], \tag{15}$$

where n_c is the number of neighbors of P_x, n_f is the number of further neighbors of P_x and n_{mf} is the number of much further neighbors of P_x. In the numerical implementation, in both pixel domain and pixon domain, h is set to 1 (12), (13) and (15). As mentioned, the *Neumann* boundary condition is used [12].

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

$$\left|P_x^{n+1}\right| = \left|P_x^n\right| + \frac{\Delta t}{f(n_c, n_f, n_{mf})} d_x^n, \tag{16}$$

where $f(n_c, n_f, n_{mf})$ depends on the number of neighboring pixons [30]. In this paper, $\Delta t / f(n_c, n_f, n_{mf}) = 0.125$. This equation is the pixonal form of (10) which is an approximation of (9). This algorithm is iteratively repeated from (13) to (16) and in each iteration, a stop criteria will be checked. This criterion is based on the quality of the image regarding the decrease in the level of noise. In the experimental section, this will be explained in more details.

Stage VI: if the dimension of the resulting pixonal image obtained by (16) has higher resolution than the original image, it is down sampled to the original dimensions.

4 Performance measure

In order to assess the performance of the proposed method compared to other methods, we use reference-based and non-reference-based metrics that are usually used with image processing techniques. The non-reference-based metrics are typically used for ultrasonic and *SAR* images. All the measures concerned are briefly introduced below.

4.1 Reference-based measures

We consider three reference-based metrics.

4.1.1 Figure of merit

The figure of merit is an edge preserving measure [37]:

$$\text{FOM} = \frac{1}{\max\{\hat{N}, N_{\text{ideal}}\}} \sum_{i=1}^{\hat{N}} \frac{1}{1 + d_i^2 \lambda}, \tag{17}$$

where $\hat{N}$ and N_{ideal} are the numbers of detected edge pixels in the noisy and the original image, respectively. d_i is the Euclidean distance between the ith detected edge pixel and the nearest original edge pixel, and λ is a constant typically set to 1/9 [38]. The dynamic range of FOM is between 0 and 1. We used the *Canny* edge detector to find edges in all the assessed techniques.

4.1.2 Structural similarity

The structural similarity (SSIM) factor can be used to measure the similarity of two images [39]. This factor consists of three different metrics. Let $X = \{x_i \,|i = 1, 2, 3, \ldots, N\}$, $Y = \left\{y_j \,|j = 1, 2, 3, \ldots, N\right\}$ be the original and the noisy image, respectively. Then, SSIM is defined as:

$$\text{SSIM} = \left(\frac{\sigma_{xy}}{\sigma_x\sigma_y}\right)^{\alpha} \times \left(\frac{2\bar{X}\bar{Y}}{(\bar{X})^2 + (\bar{Y})^2}\right)^{\beta} \times \left(\frac{2\sigma_x\sigma_y}{\sigma_x^2 + \sigma_y^2}\right)^{\gamma}$$
$$= s(x, y)^{\alpha} \times l(x, y)^{\beta} \times c(x, y)^{\gamma}, \tag{18}$$

where $\alpha > 0$, $\beta > 0$ and $\gamma > 0$ are parameters. The SSIM indicates any distortion as a combination of three different factors: loss of correlation, luminance distortion and contrast distortion. The first component is the correlation coefficient between x and y, representing the degree of linear correlation between x and y, and its dynamic range is between −1 and 1.

The highest possible value 1 is obtained when $y_i = ax_i + b$ for all $i = 1, 2, \ldots, N$ where a and b are constants and $a > 0$. The second component, with a range of [0, 1], measures how close x and y are in luminance. It equals 1 if and only if $\bar{X} = \bar{Y}$. σ_x and σ_y can be considered as an estimate of the contrast in x and y. Eventually, the third component indicates how similar the contrast levels of the images are. Its value ranges between 0 and 1, where the highest value 1 is achieved if and only if $\sigma_x = \sigma_y$.

The parameters are used to adjust the relative importance of the three conditions above. In order to simplify the expression, we set $\alpha = \beta = \gamma = 1$ in this paper.

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 $i = 1, 2, 3, \ldots, N$. The lowest value of -1 occurs when $y_i = 2\bar{X} - x_i$ for all $i = 1, 2, 3, \ldots, N$. For $\alpha = \beta = \gamma = 1$, SSIM in (18) can be rewritten as:

$$\mathrm{SSIM} = \frac{4\sigma_{xy}\bar{X}\bar{Y}}{(\sigma_x^2 + \sigma_y^2)[(\bar{X})^2 + (\bar{Y})^2]}. \tag{19}$$

In practice, to use the measure described above, the image is windowed into blocks of equal size, and SSIM is then computed for each window to find the average SSIM as follows:

$$\mathrm{MSSIM} = \overline{\mathrm{SSIM}}(X, Y) = \frac{1}{M}\sum_{j=1}^{M} \mathrm{SSIM}(x_j, y_j), \tag{20}$$

where X and Y are the original and the de-noised images respectively, M is the number of local windows in the image, x_j and y_j are the image contents at the jth local window.

4.1.3 Peak signal to noise ratio (PSNR)

The PSNR is a performance metric indicating the ratio between the maximum possible power of a signal and the power of the corrupting noise that affects the fidelity of its representation. It is most commonly used as a measure of the quality of a de-noised signal or image. It is most easily defined via the mean squared error (MSE), which is defined for two $u \times v$ images X and Y as follows:

$$\mathrm{MSE} = \frac{1}{uv}\sum_{i=0}^{u-1}\sum_{j=0}^{v-1}[X(i, j) - Y(i, j)]^2 \tag{21}$$

The PSNR is defined as:

$$\begin{aligned}\mathrm{PSNR} &= 10\log_{10}\left(\frac{(\mathrm{Max}(X))^2}{\mathrm{MSE}}\right)\\ &= 20\log_{10}\left(\frac{\mathrm{Max}\,(X)}{\sqrt{\mathrm{MSE}}}\right),\end{aligned} \tag{22}$$

where $\mathrm{Max}\,(X) = 255$ is the maximum possible value of a pixel.

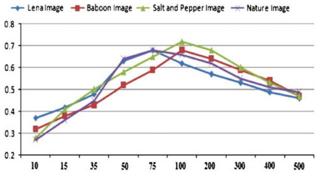


Fig. 5 MSSIM value at each iteration level of to find the optimum number of iterations

Table 1 Number of pixels and pixons

1	2	3	4	5 (%)	6 (%)
Salt and pepper	65,536	18,021	25,533	28	39
Baboon	65,536	26,455	31,168	40	47
Camera man	262,144	87,379	110,220	33	42
Indian man	262,144	115,658	121,672	44	46

Columns: *1* Images, *2* Number of pixels, The number of pixons *3* proposed method, *4 Yang*'s method, The ratio between the number of pixons and pixels, *5* proposed method, *6 Yang*'s method

4.2 Non-reference-based measures

In this paper, the non-reference-based metrics that are used to evaluate the performance of speckle reduction are noise mean value, noise standard deviation, mean square difference and deflection ratio.

4.2.1 Noise mean value (NMV)

The NMV is defined as the mean value of the de-noised image. The formula for computing NMV is:

$$\mathrm{NMV} = \frac{\sum_{i,j} Y}{M \times N}, \tag{23}$$

where $M \times N$ pixels is the size of the de-speckled image Y.

4.2.2 Noise standard deviation (NSD)

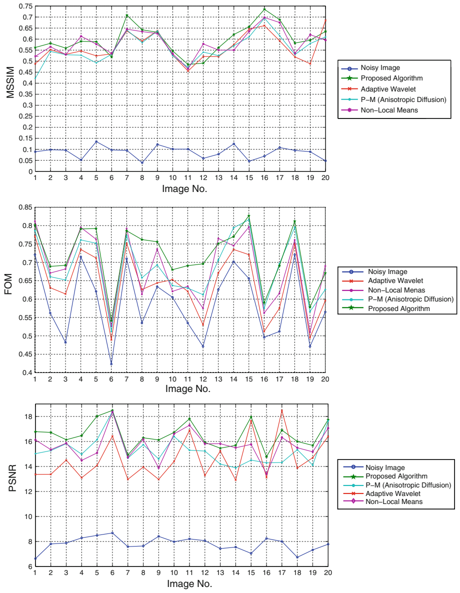
The formula for the NSD calculation is:

$$\mathrm{NSD} = \sqrt{\frac{\sum_{i,j}(Y\,(i,\,j) - \mathrm{NMV})^2}{M \times N}}. \tag{24}$$

4.2.3 Mean square difference (MSD)

MSD indicates the average squared difference between all the pixels in the original image X (with speckles) and the

Fig. 6 Results of measuring the performance of the proposed algorithms MSSIM, FOM and PSNR on 20 different images



de-speckled image Y. Lower MSD indicates a smaller difference between the original (with speckles) and the de-speckled image. The formula for the MSD calculation is:

$$\mathrm{MSD} = \frac{\sum_{i,j} \left(X\left(i,j\right) - Y\left(i,j\right)\right)^2}{M \times N}. \tag{25}$$

4.2.4 Deflection ratio (DR)

A fourth performance estimator we use in this work is the DR proposed by *H. Guo et al.* [40]. The formula for the deflection calculation is:

$$\mathrm{DR} = \frac{1}{M \times N} \sum_{i,j} \left(\frac{Y\left(i,j\right) - \mathrm{NMV}}{\mathrm{NSD}} \right). \tag{26}$$

DR should be higher at pixels with strong reflector points and lower elsewhere. In this paper, we apply these metrics to all techniques after de-speckling.

5 Experimental results

To evaluate the performance of the proposed algorithms, they have been applied on a dataset of 100 standard images taken from [41,42]. In addition, the adaptive wavelet [14], non-local algorithm [43] and anisotropic diffusion explained in Sect. 3 are implemented and their results are compared with the proposed method.

Generally, *PDE*-based approaches are iterative, and an appropriate number of iterations will enhance the image. To obtain the optimal number of iterations for this algorithm, several experiments on different images have been made. Figure 5 shows the MSSIM as a function of the number of iterations in four experiments in the range of [10, 500].

For using these techniques, the stop criterion is described as follows:

$$\mathrm{MSSIM}(n+1) < \mathrm{MSSIM}(n), \tag{27}$$

where n is the number of iterations. In the tests, the proposed method achieved maximum MSSIM in the range of [10, 500] iterations. If the number of iterations increases beyond n selected by (27), the image becomes smoother. This may damage the image.

For implementation, the threshold parameter T to extract pixons from a base image is set to 5, α in the diffusion equation (10) is 0.5 and k of the coefficient function (14) is 30.

The experiments are performed on different images and here results for four typical images are shown in Table 1. In this table, the results of the proposed method are compared with the method *Yang* and *Jiang* [29]. One advantage of using pixonal images is that the anisotropic diffusion is just applied on P pixons instead of all $M \times N$ pixels ($P \ll M \times N$). As it can be seen, the number of pixons is much lower than the number of pixels. This would facilitate the algorithm to be used in practical applications.

In these experiments, noisy images have been created by first adding Gaussian noise ($m = 64, \sigma = 400$) and then impulse noise (density = 64). The noise has been added to the images.

To implement the proposed algorithm, the noisy image is converted into a pixonal one, and then the de-noising process is performed by applying the diffusion equation. The proposed method is applied on images from [41,42], and the performance criteria, such as PSNR, FOM and MSSIM, have been used to compare the proposed method against several typical and robust methods, such as Adaptive wavelet [14], anisotropic diffusion [21,25] and non-local means [43] (these figures present the results of 20 sample images selected from the original set of 100 with a variety of texture, colors, etc.). These methods have proven very powerful for image de-noising applications during the recent years. The results of the experiments are shown in Fig. 6. As it can be seen in Fig. 6, the proposed algorithm performs better than the other methods. According to FOM, the edges of the images are preserved better. Also MSSIM indicates that the proposed approach often preserves the structure of original images better. In some cases, the non-local means algorithm performs better than the proposed method, but on the other hand, it is considerably more time consuming. The non-local mean algorithm involves all pixels of an image for de-noising one pixel, and this is why the algorithm is very slow and therefore not efficient in real time applications. Also, this method can only reduce additive noise, such as Gaussian noise, but the proposed method performs well for both additive and multiplicative noise, such as speckles.

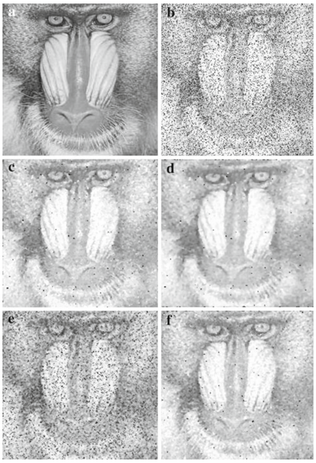


Fig. 7 Output of different methods: **a** original image, **b** noisy image [with (PSNR)$_{\mathrm{db}}$ = (1.4)$_{\mathrm{db}}$, MSSIM = 0.13, and FOM = 0.32], **c** pixon image, output of: **d** the proposed algorithm [with (PSNR)$_{\mathrm{db}}$ = (14.2)$_{\mathrm{db}}$, MSSIM = 0.62 and FOM = 0.66], **e** adaptive wavelet [with (PSNR)$_{\mathrm{db}}$ = (8.1)$_{\mathrm{db}}$, MSSIM = 0.25 and FOM = 0.36], **f** PM (anisotropic diffusion) [with (PSNR)$_{\mathrm{db}}$ = (11.2)$_{\mathrm{db}}$, MSSIM = 0.42 and FOM = 0.61]

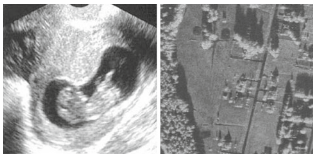


Fig. 8 An Ultrasonic and *SAR* images

These results indicate that the proposed algorithm has a better performance, compared to the other existing approaches. To judge the visual impact of this algorithm, the

Table 2 Performance of the *ADEPD* method in comparison with some other methods in ultrasonic image de-noising

	MSD	NSD	NMV	DR
Noisy image	0	42.3	82.1	1.13e−17
Median filter	158	37.5	77.4	6.5e−16
Wiener filter	168	36.3	76.1	5.7e−16
Hard thresholding	178	36.7	72.6	6.5e−16
Soft thresholding	171	35.1	75.4	8.1e−16
PM (anisotropic diffusion)	178	33.5	72.3	3.6e−15
SRAD (speckle reduction anisotropic diffusion)	185	31.7	69.8	5.7e−15
Lee filter	173	34.3	71.6	0.5e−15
Frost filter	176	32.5	72.1	1.5e−15
ADEPD method	221	31.2	68.1	8.6e−15

Table 3 Performance of the ADEPD method in comparison with some other methods in SAR image de-noising

	MSD	NSD	NMV	DR
Noisy image	0	79.5	115.2	2.21e−18
Median filter	193	73.2	110.5	4.1e−17
Wiener filter	198	71.3	108.5	5.35e−17
Hard thresholding	205	70.1	106.4	5.01e−17
Soft thresholding	212	68.5	101.8	8.9e−17
PM (anisotropic diffusion)	214	63.8	94.3	7.9e−16
SRAD (speckle reduction anisotropic diffusion)	235	56.5	91.6	9.1e−16
Lee filter	203	67.3	93.3	3.1e−16
Frost filter	215	66	90.7	8.31e−16
ADEPD method	265	52.5	83.9	12.1e−16

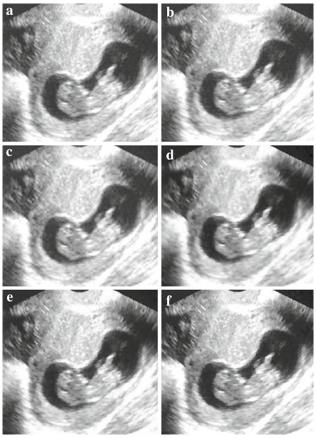


Fig. 9 The results of some de-noising methods applied to an ultrasonic image: **a** Wiener filter, **b** hard thresholding method, **c** Lee filter, **d** PM [anisotropic diffusion (iteration = 85)], **e** SRAD [speckle reduction anisotropic diffusion (iteration = 63)] and **f** ADEPD method (iteration = 33)

results from applying different techniques on the test image are shown in Fig. 7.

To evaluate the robustness of the proposed method further, the method was also applied to ultrasonic and SAR images (see Fig. 8). These images are often influenced by multiplicative noise such as speckle noise, and many research studies have focused on de-noising this type of images [41,42]. In this paper, we have compared the ADEPD method with some of the previously studied methods, such as median filter, Wiener filter, hard and soft thresholding wavelet [41], Lee filter, Frost filter [44] and speckle reduction anisotropic diffusion (SRAD) [12]. Some measures, such as NMV, NSD, MSD and DR, have been applied to show the supremacy of the ADEPD method in comparison with the others. The results are shown in Tables 2 and 3. The results indicate that the ADEPD method is capable of de-noising natural images.

Figures 9 and 10 depict some results of de-noising with the proposed method and other existing methods for ultrasonic and *SAR* images. These results show that the proposed method is robust in noisy situations. In the result obtained by Wiener filtering in Figs. 9a and 10a, the speckle noise is reduced well and structures are enhanced. But some details are lost and some are over-enhanced. Meanwhile, in the result given by the wavelet method in Figs. 9b and 10b, the speckle noise is reduced relatively well, but structures are blurred and some visible artifacts are introduced. Although the *Lee* filter achieves a good speckle noise suppression performance, it loses many of the signal details and the resulting image is blurred (Figs. 9c, 10c). In addition, the PM method (Figs. 9d, 10d) and SRAD method (Figs. 9d, 10d) results in good noise reduction, but the output image suffers from smoothing effect, especially at the edges, while the result of the proposed algorithm given in Figs. 9d and 10d shows that speckle noise is efficiently reduced and structures are enhanced with almost no loss or noticeable artifacts.

To evaluate the computational cost of the proposed method in comparison with the PM method, these methods are applied to 100 different images (from [41,42]). In both cases,

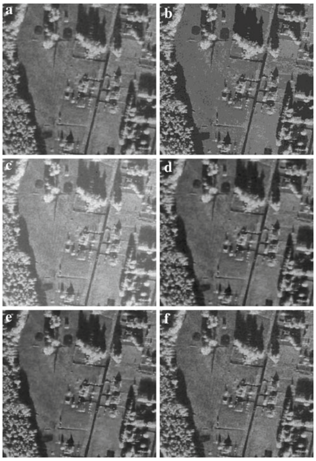


Fig. 10 The results of some de-noising methods applied to a *SAR*: **a** Wiener filter, **b** hard thresholding method, **c** *Lee* filter, **d** PM [anisotropic diffusion (iteration = 63)], **e** SRAD [speckle reduction anisotropic diffusion (iteration = 55)] and **f** ADEPD method (iteration = 48)

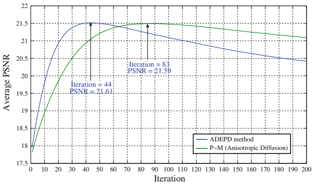


Fig. 11 The results of comparison of the proposed method and PM (anisotropic diffusion) method

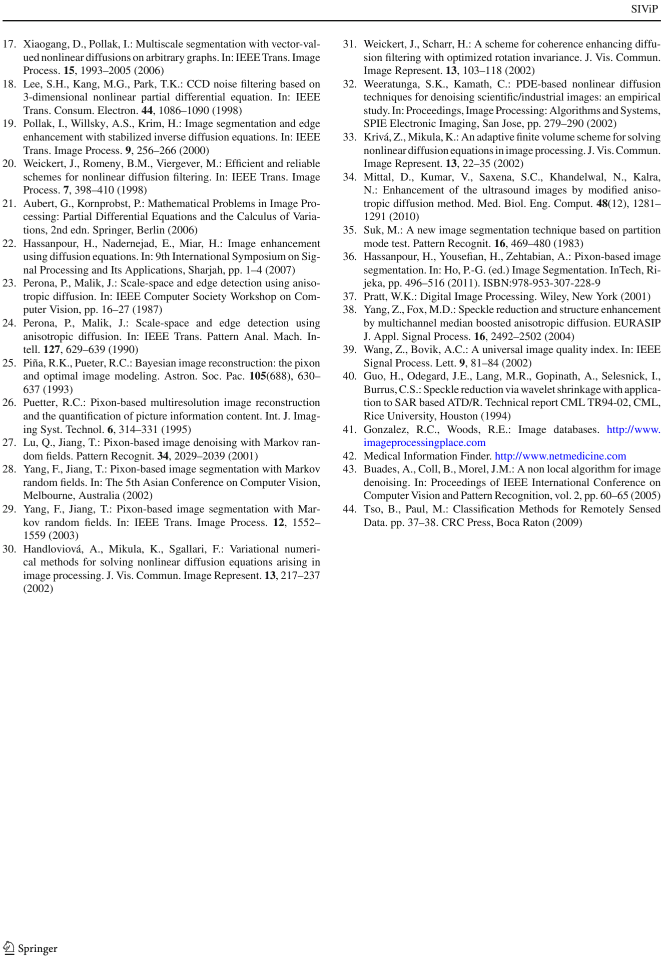
200 iterations are used. The average PSNR versus the number of iterations is depicted in Fig. 11. This figure shows that the proposed method achieved the maximum PSNR at iteration 44 rather than PM method which reached its peak at iteration 83.

6 Conclusion

In this paper, a new algorithm for image de-noising has been proposed. The method uses diffusion equations and a pixonal model of the image. The proposed algorithm enhances the image while preserving the edges in the image. The comparative results indicate the effectiveness of the proposed method. The experimental results show that, in comparison with other existing methods, the proposed algorithm has a better performance.

References

1. Bouzari, H.: An improved regularization method for artifact rejection in image super-resolution. Signal Image Video Process. **6**(1), 125–140 (2012)
2. Min, T.H., Park, R.H., Chang, S.K.: Noise reduction in high dynamic range images. Signal Image Video Process **5**(3), 315–328 (2011)
3. Santis, A.D., Iacoviello, D.: A discrete level set approach to image segmentation. Signal Image Video Process. **1**(4), 303–320 (2007)
4. Gonzalez, R.C., Woods, R.E.: Digital Image Processing. Prentice Hall, Englewood Cliffs, NJ (2008)
5. Zhong, J., Ning, R., Conover, D.: Image denoising based on multiscale singularity detection for cone Beam CT breast imaging. In: IEEE Trans. Med. Imaging **23**, 696–703 (2004)
6. Mielikäinen, T., Ravantti, J.: Sinogram denoising of cryo-electron microscopy images. In: International Conference on Computational Science and its Applications, Singapore, pp. 9–12 (2005)
7. Senel, H.G., Peteres, R.A., Dawant, B.: Topological median filters. In: IEEE Trans. Image Process. **11**, 89–104 (2002)
8. Koschan, A.F., Abidi, M.A.: A comparison of median filter techniques for noise removal in color images. In: Proceedings of 7th German Workshop on Color Image Processing, Erlangen, Germany, pp. 69–79 (2001)
9. Wang, Z., Qu, C., Cui, L.: Denoising images using Wiener filter in directionalet domain. In: Proceedings of International Conference on Computational Intelligence for Modeling Control and Automation. NSW, Sydney (2006)
10. Fodor, I.E., Kamath, C.: On denoising images using wavelet-based statistical techniques. Technical report UCRLJC-142357, Lawrence Livermore National Laboratory (2001)
11. Jansen, M.: Noise Reduction by Wavelet Thresholding. Springer, New York (2001)
12. Yu, Y., Acton, S.T.: (2002) Speckle reducing anisotropic diffusion. In: IEEE Trans. Image Process. **11**(11), 1260–1270
13. Chang, S.G., Yu, B., Vetterli, M.: Spatially adaptive wavelet thresholding with context modeling for image denoising. In: IEEE Trans. Image Process. **9**(9), 1522–1531 (2000)
14. Chang, S.G., Yu, B., Vetterli, M.: Adaptive wavelet thresholding for image denoising and compression. In: IEEE Trans. Image Process. **9**(9), 1532–1545 (2000)
15. Smigiel, E., Alby, E., Grussenmeyer, P.: Terrestrial Laser Scanner Data Denoising by Range Image Processing for Small-Sized Objects. ISPRS Congress Beijing (2008)
16. Catte, F., Lions, P.L., Morel, J.M., Coll, T.: Image selective smoothing and edge detection by nonlinear diffusion. SIAM J. Numer. Anal. **29**(1), 182–193 (1992)



17. Xiaogang, D., Pollak, I.: Multiscale segmentation with vector-valued nonlinear diffusions on arbitrary graphs. In: IEEE Trans. Image Process. **15**, 1993–2005 (2006)
18. Lee, S.H., Kang, M.G., Park, T.K.: CCD noise filtering based on 3-dimensional nonlinear partial differential equation. In: IEEE Trans. Consum. Electron. **44**, 1086–1090 (1998)
19. Pollak, I., Willsky, A.S., Krim, H.: Image segmentation and edge enhancement with stabilized inverse diffusion equations. In: IEEE Trans. Image Process. **9**, 256–266 (2000)
20. Weickert, J., Romeny, B.M., Viergever, M.: Efficient and reliable schemes for nonlinear diffusion filtering. In: IEEE Trans. Image Process. **7**, 398–410 (1998)
21. Aubert, G., Kornprobst, P.: Mathematical Problems in Image Processing: Partial Differential Equations and the Calculus of Variations, 2nd edn. Springer, Berlin (2006)
22. Hassanpour, H., Nadernejad, E., Miar, H.: Image enhancement using diffusion equations. In: 9th International Symposium on Signal Processing and Its Applications, Sharjah, pp. 1–4 (2007)
23. Perona, P., Malik, J.: Scale-space and edge detection using anisotropic diffusion. In: IEEE Computer Society Workshop on Computer Vision, pp. 16–27 (1987)
24. Perona, P., Malik, J.: Scale-space and edge detection using anisotropic diffusion. In: IEEE Trans. Pattern Anal. Mach. Intell. **127**, 629–639 (1990)
25. Piña, R.K., Pueter, R.C.: Bayesian image reconstruction: the pixon and optimal image modeling. Astron. Soc. Pac. **105**(688), 630–637 (1993)
26. Puetter, R.C.: Pixon-based multiresolution image reconstruction and the quantification of picture information content. Int. J. Imaging Syst. Technol. **6**, 314–331 (1995)
27. Lu, Q., Jiang, T.: Pixon-based image denoising with Markov random fields. Pattern Recognit. **34**, 2029–2039 (2001)
28. Yang, F., Jiang, T.: Pixon-based image segmentation with Markov random fields. In: The 5th Asian Conference on Computer Vision, Melbourne, Australia (2002)
29. Yang, F., Jiang, T.: Pixon-based image segmentation with Markov random fields. In: IEEE Trans. Image Process. **12**, 1552–1559 (2003)
30. Handloviová, A., Mikula, K., Sgallari, F.: Variational numerical methods for solving nonlinear diffusion equations arising in image processing. J. Vis. Commun. Image Represent. **13**, 217–237 (2002)
31. Weickert, J., Scharr, H.: A scheme for coherence enhancing diffusion filtering with optimized rotation invariance. J. Vis. Commun. Image Represent. **13**, 103–118 (2002)
32. Weeratunga, S.K., Kamath, C.: PDE-based nonlinear diffusion techniques for denoising scientific/industrial images: an empirical study. In: Proceedings, Image Processing: Algorithms and Systems, SPIE Electronic Imaging, San Jose, pp. 279–290 (2002)
33. Krivá, Z., Mikula, K.: An adaptive finite volume scheme for solving nonlinear diffusion equations in image processing. J. Vis. Commun. Image Represent. **13**, 22–35 (2002)
34. Mittal, D., Kumar, V., Saxena, S.C., Khandelwal, N., Kalra, N.: Enhancement of the ultrasound images by modified anisotropic diffusion method. Med. Biol. Eng. Comput. **48**(12), 1281–1291 (2010)
35. Suk, M.: A new image segmentation technique based on partition mode test. Pattern Recognit. **16**, 469–480 (1983)
36. Hassanpour, H., Yousefian, H., Zehtabian, A.: Pixon-based image segmentation. In: Ho, P.-G. (ed.) Image Segmentation. InTech, Rijeka, pp. 496–516 (2011). ISBN:978-953-307-228-9
37. Pratt, W.K.: Digital Image Processing. Wiley, New York (2001)
38. Yang, Z., Fox, M.D.: Speckle reduction and structure enhancement by multichannel median boosted anisotropic diffusion. EURASIP J. Appl. Signal Process. **16**, 2492–2502 (2004)
39. Wang, Z., Bovik, A.C.: A universal image quality index. In: IEEE Signal Process. Lett. **9**, 81–84 (2002)
40. Guo, H., Odegard, J.E., Lang, M.R., Gopinath, A., Selesnick, I., Burrus, C.S.: Speckle reduction via wavelet shrinkage with application to SAR based ATD/R. Technical report CML TR94-02, CML, Rice University, Houston (1994)
41. Gonzalez, R.C., Woods, R.E.: Image databases. http://www.imageprocessingplace.com
42. Medical Information Finder. http://www.netmedicine.com
43. Buades, A., Coll, B., Morel, J.M.: A non local algorithm for image denoising. In: Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition, vol. 2, pp. 60–65 (2005)
44. Tso, B., Paul, M.: Classification Methods for Remotely Sensed Data. pp. 37–38. CRC Press, Boca Raton (2009)

A.8 Improving Performance of Wavelet-based Image Denoising Algorithm using Complex Diffusion Process

E. Nadernejad, S. Sharifzadeh, and J. Korhonen, “Improving Performance of Wavelet-based Image Denoising Algorithm Using Complex Diffusion Process”, *Imaging Science Journal*, vol. 60, no. 4, pp. 208–218, 2012

Reference: [8]

Improving performance of wavelet-based image de-noising algorithm using complex diffusion process

E Nadernejad*[a], S Sharifzadeh[b] and **J Korhonen[a]**

[a]Department of Photonics Engineering, Ørsteds Plads, Building 343, Technical University of Denmark, 2800 Kgs., Lyngby, Denmark

[b]Department of Microelectronics and Electronic Systems, Centre for Hardware–Software Prototypes and Solutions, University Autonoma de Barcelona, School of Engineering, Campus UAB, 08193 Bellaterra, Barcelona, Spain

Abstract: Image enhancement and de-noising is an essential pre-processing step in many image processing algorithms. In any image de-noising algorithm, the main concern is to keep the interesting structures of the image. Such interesting structures often correspond to the discontinuities (edges). In this paper, we present a new algorithm for image noise reduction based on the combination of complex diffusion process and wavelet thresholding. In the existing wavelet thresholding methods, the noise reduction is limited, because the approximate coefficients containing the main information of the image are kept unchanged. Since noise affects both the approximate and detail coefficients, the proposed algorithm for noise reduction applies the complex diffusion process on the approximation band in order to alleviate the deficiency of the existing wavelet thresholding methods. The algorithm has been examined using a variety of standard images and its performance has been compared against several de-noising algorithms known from the prior art. Experimental results show that the proposed algorithm preserves the edges better and in most cases, improves the measured visual quality of the de-noised images in comparison to the existing methods known from the literature. The improvement is obtained without excessive computational cost, and the algorithm works well on a wide range of different types of noise.

Keywords: complex diffusion process, wavelet transform, thresholding, image de-noising

1 INTRODUCTION

In many image processing applications, such as medical imaging and astronomical studies, noise reduction is used as a pre-processing step.[1,2] An ideal algorithm for de-noising removes the undesired noise artefacts efficiently while preserving all the informative structures in the image as accurately as possible. It is a real challenge to obtain these two contradictory goals simultaneously, and this challenge has motivated our work reported in this paper.

Wavelet thresholding is a method that has been used in several related studies focusing in noise reduction.[3,4] In this approach, a threshold is applied on wavelet coefficients. This method preserves the edges of the image. However, it may destroy texture in an image, because in this approach, pixel values in a region with small variations are transformed into a constant value.

Since images are considered as a non-stationary process,[5] adaptive filters are mostly applicable in image noise reduction. Some of these filters can be designed using partial differential equations (PDEs).

The MS was accepted for publication on 23 May 2011.

** Corresponding author: Ehsan Nadernejad, Department of Photonics Engineering, Ørsteds Plads, Building 343, Technical University of Denmark, 2800 Kgs., Lyngby, Denmark; email: ehsa@fotonik.dtu.dk*

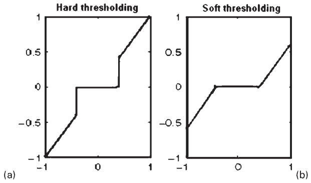
IMAG 145 © RPS 2012 DOI: 10.1179/1743131X11Y.0000000024

In many research studies, PDE is employed as an efficient method for noise reduction. These equations are employed to overcome the edge smoothing effect that exists in mean and Gaussian filters.[6] Two equations have been introduced in Refs. 7 and 8 for non-linear diffusion process. These equations apply low-pass filters on the image, smoothing the regions with a low gradient and preserving the high gradient regions, such as edges. The existing diffusion-based methods suggest the use of the side neighbours.[9] Research studies show that these methods may also lead to smoothing of the edges;[10,11] however, this effect is smaller than if mean and Gaussian filters are used.[12]

The concept of complex diffusion in image processing was introduced by Gilboa *et al.*[13] as an alternative to second order anisotropic diffusion, which introduces blocky effects in images. This blocky effect is inherent in the nature of ordinary second order equations; it can be avoided by using complex diffusion. Complex diffusion is derived by combining the standard diffusion equation with the free Schrödinger equation.[14] Even though it violates the maximum–minimum principle, it preserves other desirable mathematical and perceptual properties. When an image is processed with complex diffusion, we will get low-frequency components (plateaus) of the image in the real plane and high-frequency components (edges) in the imaginary plane.[11] The components in the real and imaginary planes are almost equivalent to that of the image convolved with a Gaussian and Laplacian of Gaussian at various scales, respectively.

In this paper, a new method based on complex diffusion equation and wavelet thresholding for image enhancement is presented. One of the important characteristics of the complex diffusion process is that when complex diffusion coefficients are close to the real axis, the imaginary part in these equations can be considered as edge detector (smoothed second derivative with respect to time). In the proposed approach, the image is initially decomposed using wavelet transform. Then, the sub-bands containing the detail coefficients are thresholded using wavelet thresholding method, and the complex diffusion is applied on the approximate coefficients.

The rest of the paper is organised as follows. The existing image de-noising techniques are reviewed in Section 2. Section 3 describes the proposed de-noising algorithm. In Section 4, the performance criteria used in this paper is presented. Experimental results are given in Section 5 for several test images and three noise models. Finally, the conclusions are drawn in Section 6.



1 Thresholding functions for the noise cancelling operations

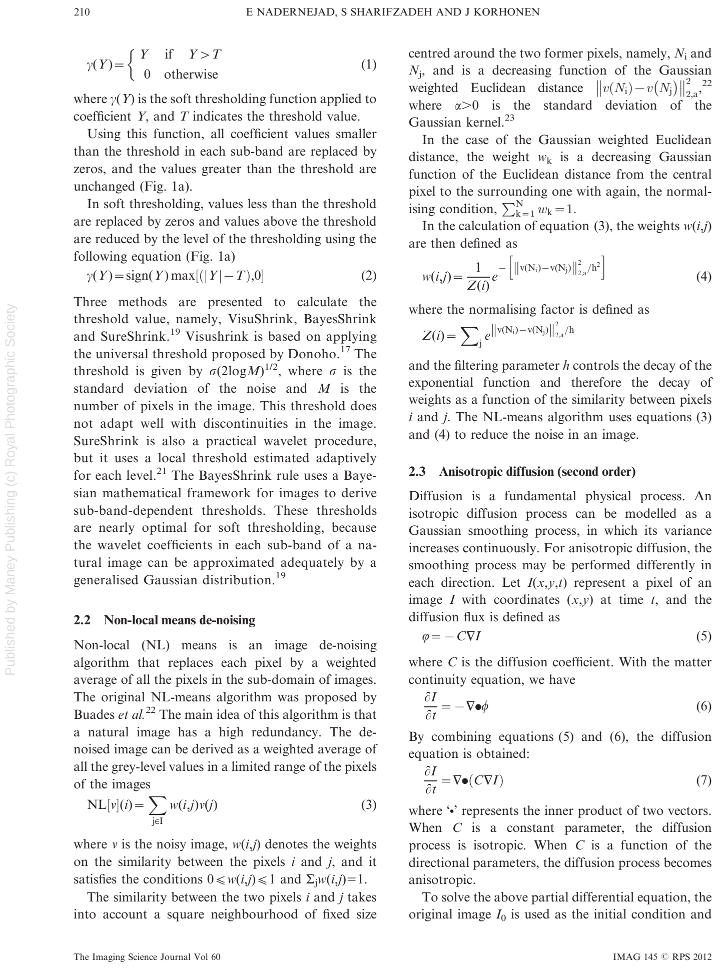
2 REVIEWING THE EXISTING IMAGE DE-NOISING TECHNIQUES

There are many image de-noising approaches in literature. Some of these methods are briefly described here. These methods are used in this paper for the comparison analysis of the proposed method.

2.1 Wavelet thresholding method

In recent years, many researchers have performed wavelet thresholding for image de-noising.[15–21] Thresholding is a simple non-linear technique operating on the wavelet coefficients. In this technique, each coefficient is thresholded by comparing it to a threshold value. If the coefficient is smaller than the threshold, it is set to zero; otherwise, it is kept or modified as defined in the used thresholding method. The motivation is that as the wavelet transform is good at energy compaction, the small coefficients represent noise and large coefficient indicate the important signal features.[17,18] The small coefficients can be thresholded without affecting the significant features of the image. By applying inverse wavelet transform on the thresholded coefficients, a reconstructed image with lower noise is obtained.

In applications, a soft thresholding method is more desirable than the hard thresholding, because it gives visually more pleasant images. The hard thresholding may lead to discontinuous and abrupt artefacts in the de-noised images, especially when the noise energy is significant. The brief description of the hard thresholding is as follows



$$\gamma(Y)=\begin{cases} Y & \text{if} \quad Y>T \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

where $\gamma(Y)$ is the soft thresholding function applied to coefficient Y, and T indicates the threshold value.

Using this function, all coefficient values smaller than the threshold in each sub-band are replaced by zeros, and the values greater than the threshold are unchanged (Fig. 1a).

In soft thresholding, values less than the threshold are replaced by zeros and values above the threshold are reduced by the level of the thresholding using the following equation (Fig. 1a)

$$\gamma(Y)=\mathrm{sign}(Y)\max[(|Y|-T),0] \tag{2}$$

Three methods are presented to calculate the threshold value, namely, VisuShrink, BayesShrink and SureShrink.[19] Visushrink is based on applying the universal threshold proposed by Donoho.[17] The threshold is given by $\sigma(2\log M)^{1/2}$, where σ is the standard deviation of the noise and M is the number of pixels in the image. This threshold does not adapt well with discontinuities in the image. SureShrink is also a practical wavelet procedure, but it uses a local threshold estimated adaptively for each level.[21] The BayesShrink rule uses a Bayesian mathematical framework for images to derive sub-band-dependent thresholds. These thresholds are nearly optimal for soft thresholding, because the wavelet coefficients in each sub-band of a natural image can be approximated adequately by a generalised Gaussian distribution.[19]

2.2 Non-local means de-noising

Non-local (NL) means is an image de-noising algorithm that replaces each pixel by a weighted average of all the pixels in the sub-domain of images. The original NL-means algorithm was proposed by Buades *et al.*[22] The main idea of this algorithm is that a natural image has a high redundancy. The de-noised image can be derived as a weighted average of all the grey-level values in a limited range of the pixels of the images

$$\mathrm{NL}[v](i)=\sum_{j\in I} w(i,j)v(j) \tag{3}$$

where v is the noisy image, $w(i,j)$ denotes the weights on the similarity between the pixels i and j, and it satisfies the conditions $0\leqslant w(i,j)\leqslant 1$ and $\Sigma_j w(i,j)=1$.

The similarity between the two pixels i and j takes into account a square neighbourhood of fixed size centred around the two former pixels, namely, N_i and N_j, and is a decreasing function of the Gaussian weighted Euclidean distance $\left\|v(N_i)-v(N_j)\right\|_{2,a}^2$,[22] where $\alpha>0$ is the standard deviation of the Gaussian kernel.[23]

In the case of the Gaussian weighted Euclidean distance, the weight w_k is a decreasing Gaussian function of the Euclidean distance from the central pixel to the surrounding one with again, the normalising condition, $\sum_{k=1}^{N} w_k=1$.

In the calculation of equation (3), the weights $w(i,j)$ are then defined as

$$w(i,j)=\frac{1}{Z(i)}e^{-\left[\left\|v(N_i)-v(N_j)\right\|_{2,a}^2/h^2\right]} \tag{4}$$

where the normalising factor is defined as

$$Z(i)=\sum\nolimits_j e^{\left\|v(N_i)-v(N_j)\right\|_{2,a}^2/h}$$

and the filtering parameter h controls the decay of the exponential function and therefore the decay of weights as a function of the similarity between pixels i and j. The NL-means algorithm uses equations (3) and (4) to reduce the noise in an image.

2.3 Anisotropic diffusion (second order)

Diffusion is a fundamental physical process. An isotropic diffusion process can be modelled as a Gaussian smoothing process, in which its variance increases continuously. For anisotropic diffusion, the smoothing process may be performed differently in each direction. Let $I(x,y,t)$ represent a pixel of an image I with coordinates (x,y) at time t, and the diffusion flux is defined as

$$\varphi=-C\nabla I \tag{5}$$

where C is the diffusion coefficient. With the matter continuity equation, we have

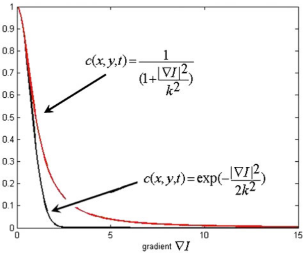
$$\frac{\partial I}{\partial t}=-\nabla\bullet\phi \tag{6}$$

By combining equations (5) and (6), the diffusion equation is obtained:

$$\frac{\partial I}{\partial t}=\nabla\bullet(C\nabla I) \tag{7}$$

where '•' represents the inner product of two vectors. When C is a constant parameter, the diffusion process is isotropic. When C is a function of the directional parameters, the diffusion process becomes anisotropic.

To solve the above partial differential equation, the original image I_0 is used as the initial condition and



2 Diffusion coefficient curve versus gradient variations

the Neumann boundary condition is applied to the image borders

$$I(x,y,t)_{\mathrm{t}=0}=I_0 \qquad (8)$$
$$\partial_{\mathrm{n}} I=0$$

Discrete form of diffusion equation is as follows

$$\frac{\partial I(x,y,t)}{\partial t}=\nabla\bullet[C(x,y,t)\nabla I(x,y,t)] \qquad (9)$$
$$I(x,y,0)=I_0(x,y)$$

where C is diffusion coefficient and I is the original image in each time scale t.

Perona–Malik (PM)[7,8] suggested the two well-known diffusion coefficients as follows

$$C(s)=\frac{1}{1+(s/k)^2} \qquad (10)$$

$$C(s)=\exp\left[-\left(\frac{s}{k}\right)^2\right] \qquad (11)$$

where $s=|\nabla I|$. By using diffusion equations, the diffusion process will be prominent when the magnitude of the local gradient is low, and will be restricted when the magnitude of the local gradient is high, as shown in Fig. 2.

In both equations (10) and (11), the parameter k is a factor controlling the diffusion process. Both equations provide perceptually similar results, but equation (10) emphasises noise removal, while equation (11) emphasises high-contrast preservation.

In some cases, the PM method can misinterpret noise as an edge and enhance it, creating false edges. Therefore, Catte *et al.*[24] changed $s=|\nabla I|$ in the PM diffusion function to

$$s=|\nabla G_\sigma * I| \qquad (12)$$

where G_σ is Gaussian smoothing kernel and '*' is the convolution operator. In this approach, $|\nabla G_\sigma * I|$ is used to estimate the local gradient more accurately than the noise sensitive$|\nabla I|$.

2.4 Complex diffusion process

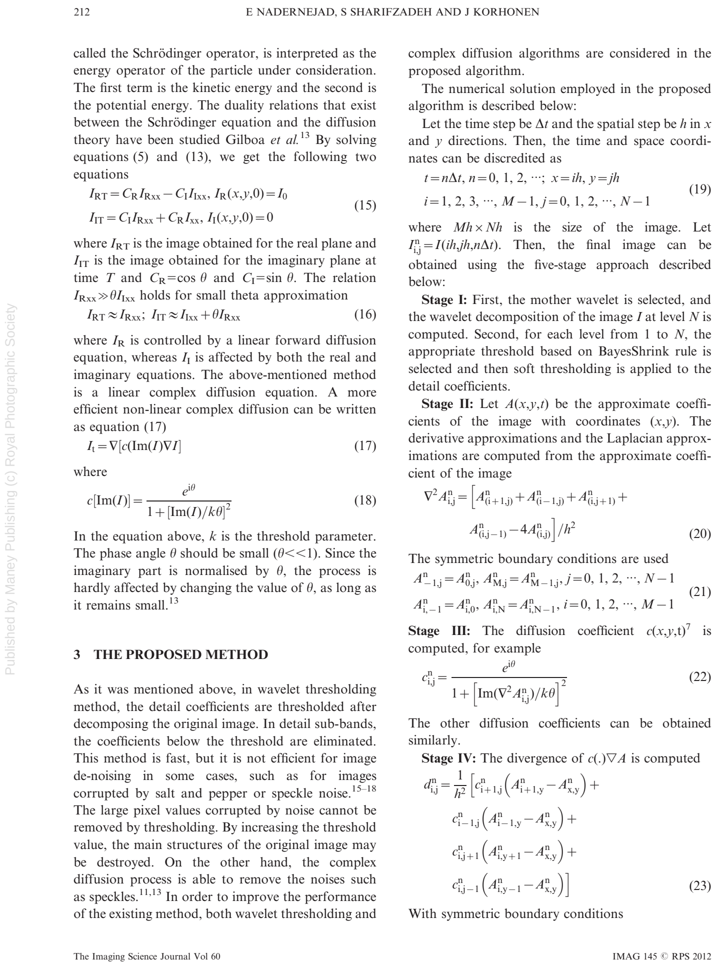
Complex diffusion is a relatively new method and can be applied to image de-noising. It is a generalisation of diffusion and free Schrödinger equations. It was realised in various areas of physics and engineering that extending the analysis from the real axis to the complex domain is very helpful, even though the variables and/or quantities of interest are real. Analysis of linear complex diffusion shows that the generalised diffusion has properties of both forward and inverse diffusion.[13]

In 1931, Schrödinger proposed the possibility of using diffusion theory as a starting point to derive the equations of quantum theory. This idea was developed by Fuerth, who indicated that the Schrödinger equation could be derived from the diffusion equation by introducing a relation between the diffusion coefficient and Planck's constant, and stipulating that the probability amplitude of quantum theory should be given by the resulting differential equation.[14] It has been the goal of a variety of subsequent approaches to derive the probabilistic equations of quantum mechanics from equations involving probabilistic or stochastic processes. Complex diffusion-type processes are encountered in quantum physics and in electro optics. The time-dependent Schrödinger equation is the fundamental equation of quantum mechanics.[13] In the simplest case for a particle without spin in an external field, it has the form

$$i\eta\frac{\partial\psi}{\partial t}=-\frac{\eta^2}{2m}\Delta\psi+V(x)\psi \qquad (13)$$

where $\psi=\psi(x,t)$ is the wave function of a quantum particle, m is the mass of the particle, η is Planck's constant, $V(x)$ is the external field potential, Δ is the Laplacian and $i=(-1)^{1/2}$. With the initial condition $\psi|_{\mathrm{t}=0}=\psi_0(x)$, requiring that $\psi(0,t)L_2$ for each fixed t, the solution is $\psi(0,t)=e^{(-\mathrm{i/h})\mathrm{tH}}\psi_0$ where the exponent is shorthand for the corresponding power series, and the higher order terms are defined recursively by $H^{\mathrm{n}}\psi=H(H^{\mathrm{n}-1}\psi)$. The operator

$$H=-\frac{\eta^2}{2m}\Delta+V(x) \qquad (14)$$



called the Schrödinger operator, is interpreted as the energy operator of the particle under consideration. The first term is the kinetic energy and the second is the potential energy. The duality relations that exist between the Schrödinger equation and the diffusion theory have been studied Gilboa *et al.*[13] By solving equations (5) and (13), we get the following two equations

$$I_{\mathrm{RT}}=C_{\mathrm{R}}I_{\mathrm{Rxx}}-C_{\mathrm{I}}I_{\mathrm{Ixx}},\ I_{\mathrm{R}}(x,y,0)=I_0$$
$$I_{\mathrm{IT}}=C_{\mathrm{I}}I_{\mathrm{Rxx}}+C_{\mathrm{R}}I_{\mathrm{xx}},\ I_{\mathrm{I}}(x,y,0)=0 \tag{15}$$

where I_{RT} is the image obtained for the real plane and I_{IT} is the image obtained for the imaginary plane at time T and $C_{\mathrm{R}}=\cos\theta$ and $C_{\mathrm{I}}=\sin\theta$. The relation $I_{\mathrm{Rxx}}\gg\theta I_{\mathrm{Ixx}}$ holds for small theta approximation

$$I_{\mathrm{RT}}\approx I_{\mathrm{Rxx}};\ I_{\mathrm{IT}}\approx I_{\mathrm{Ixx}}+\theta I_{\mathrm{Rxx}} \tag{16}$$

where I_{R} is controlled by a linear forward diffusion equation, whereas I_{I} is affected by both the real and imaginary equations. The above-mentioned method is a linear complex diffusion equation. A more efficient non-linear complex diffusion can be written as equation (17)

$$I_{\mathrm{t}}=\nabla[c(\mathrm{Im}(I)\nabla I] \tag{17}$$

where

$$c[\mathrm{Im}(I)]=\frac{e^{\mathrm{i}\theta}}{1+[\mathrm{Im}(I)/k\theta]^2} \tag{18}$$

In the equation above, k is the threshold parameter. The phase angle θ should be small ($\theta<<1$). Since the imaginary part is normalised by θ, the process is hardly affected by changing the value of θ, as long as it remains small.[13]

3 THE PROPOSED METHOD

As it was mentioned above, in wavelet thresholding method, the detail coefficients are thresholded after decomposing the original image. In detail sub-bands, the coefficients below the threshold are eliminated. This method is fast, but it is not efficient for image de-noising in some cases, such as for images corrupted by salt and pepper or speckle noise.[15–18] The large pixel values corrupted by noise cannot be removed by thresholding. By increasing the threshold value, the main structures of the original image may be destroyed. On the other hand, the complex diffusion process is able to remove the noises such as speckles.[11,13] In order to improve the performance of the existing method, both wavelet thresholding and complex diffusion algorithms are considered in the proposed algorithm.

The numerical solution employed in the proposed algorithm is described below:

Let the time step be Δt and the spatial step be h in x and y directions. Then, the time and space coordinates can be discredited as

$$t=n\Delta t,\ n=0,\ 1,\ 2,\ \cdots;\ x=ih,\ y=jh$$
$$i=1,\ 2,\ 3,\ \cdots,\ M-1,\ j=0,\ 1,\ 2,\ \cdots,\ N-1 \tag{19}$$

where $Mh\times Nh$ is the size of the image. Let $I^{\mathrm{n}}_{\mathrm{i,j}}=I(ih,jh,n\Delta t)$. Then, the final image can be obtained using the five-stage approach described below:

Stage I: First, the mother wavelet is selected, and the wavelet decomposition of the image I at level N is computed. Second, for each level from 1 to N, the appropriate threshold based on BayesShrink rule is selected and then soft thresholding is applied to the detail coefficients.

Stage II: Let $A(x,y,t)$ be the approximate coefficients of the image with coordinates (x,y). The derivative approximations and the Laplacian approximations are computed from the approximate coefficient of the image

$$\nabla^2A^{\mathrm{n}}_{\mathrm{i,j}}=\Big[A^{\mathrm{n}}_{(\mathrm{i+1,j})}+A^{\mathrm{n}}_{(\mathrm{i-1,j})}+A^{\mathrm{n}}_{(\mathrm{i,j+1})}+$$
$$A^{\mathrm{n}}_{(\mathrm{i,j-1})}-4A^{\mathrm{n}}_{(\mathrm{i,j})}\Big]/h^2 \tag{20}$$

The symmetric boundary conditions are used

$$A^{\mathrm{n}}_{-1,\mathrm{j}}=A^{\mathrm{n}}_{0,\mathrm{j}},\ A^{\mathrm{n}}_{\mathrm{M,j}}=A^{\mathrm{n}}_{\mathrm{M-1,j}},\ j=0,\ 1,\ 2,\ \cdots,\ N-1$$
$$A^{\mathrm{n}}_{\mathrm{i},-1}=A^{\mathrm{n}}_{\mathrm{i},0},\ A^{\mathrm{n}}_{\mathrm{i,N}}=A^{\mathrm{n}}_{\mathrm{i,N-1}},\ i=0,\ 1,\ 2,\ \cdots,\ M-1 \tag{21}$$

Stage III: The diffusion coefficient $c(x,y,\mathrm{t})$[7] is computed, for example

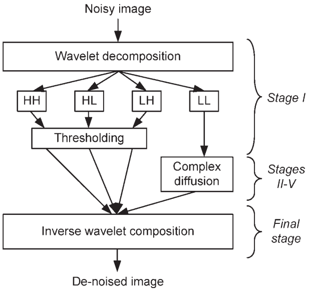
$$c^{\mathrm{n}}_{\mathrm{i,j}}=\frac{e^{\mathrm{i}\theta}}{1+\Big[\mathrm{Im}(\nabla^2A^{\mathrm{n}}_{\mathrm{i,j}})/k\theta\Big]^2} \tag{22}$$

The other diffusion coefficients can be obtained similarly.

Stage IV: The divergence of $c(.)\nabla A$ is computed

$$d^{\mathrm{n}}_{\mathrm{i,j}}=\frac{1}{h^2}\Big[c^{\mathrm{n}}_{\mathrm{i+1,j}}\Big(A^{\mathrm{n}}_{\mathrm{i+1,y}}-A^{\mathrm{n}}_{\mathrm{x,y}}\Big)+$$
$$c^{\mathrm{n}}_{\mathrm{i-1,j}}\Big(A^{\mathrm{n}}_{\mathrm{i-1,y}}-A^{\mathrm{n}}_{\mathrm{x,y}}\Big)+$$
$$c^{\mathrm{n}}_{\mathrm{i,j+1}}\Big(A^{\mathrm{n}}_{\mathrm{i,y+1}}-A^{\mathrm{n}}_{\mathrm{x,y}}\Big)+$$
$$c^{\mathrm{n}}_{\mathrm{i,j-1}}\Big(A^{\mathrm{n}}_{\mathrm{i,y-1}}-A^{\mathrm{n}}_{\mathrm{x,y}}\Big)\Big] \tag{23}$$

With symmetric boundary conditions



3 Structure of the proposed method

$$\begin{aligned} &d^{n}_{-1,j}=d^{n}_{0,j},\ d^{n}_{M,j}=d^{n}_{M-1,j},\ j=0,1,2,\cdots,N-1 \\ &d^{n}_{i,-1}=d^{n}_{i,0},\ d^{n}_{i,N}=d^{n}_{i,N-1},\ i=0,1,2,\cdots,M-1 \end{aligned} \qquad (24)$$

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

$$A^{n+1}_{i,j}=A^{n}_{i,j}+\frac{\Delta t}{16}d^{n}_{i,j} \qquad (25)$$

This equation is equivalent to equation (5) (in this paper: $\Delta t/16\approx 0.125$).

It should be noted that in any PDE-based method, the number of iterations is crucial. In other words, too many iterations may cause the pixel value to exceed the grey level value, relocating the edges and blurring the image; on the other hand, with too few iterations. the image cannot be de-noised efficiently.

Final Stage: The inverse wavelet transform is applied on the sub-bands to obtain the de-noised image. The proposed method is illustrated in Fig. 3.

4 PERFORMANCE MEASURE

In this paper, three well-known metrics are used to evaluate the algorithms. They are introduced here briefly.

4.1 Figure of merit metric

The figure of merit (FOM) is an edge preserving measure that is defined as below[25]

$$\mathrm{FOM}=\frac{1}{\max\left[\hat{N},N_{\mathrm{ideal}}\right]}\sum_{i=1}^{\hat{N}}\frac{1}{1+d_i^2\lambda} \qquad (26)$$

where $\hat{N}$ and N_{ideal} are the numbers of detected and original edge pixels, respectively, d_i is the Euclidean distance between the ith detected edge pixel and the nearest original edge pixel, and λ is a constant typically set to $^1/_9$. The dynamic range of FOM is between the processed image and the ideal image. We used the Canny edge detector[26] to find the edge in all processed results.

4.2 Structural similarity metric

The structural similarity metric (SSIM) proposed by Wang and Bovik[27] consists of three different metrics. Let $x\{x_i|i=1, 2, 3, \ldots, N\}$ and $y=\{y_i|i=1, 2, 3, \ldots, N\}$ be the original and the test images, respectively. The proposed quality index is defined as

$$\mathrm{SSIM}=Q=\frac{4\sigma_{XY}\overline{X}\,\overline{Y}}{\left(\sigma_X^2+\sigma_Y^2\right)\left[\left(\overline{X}\right)^2+\left(\overline{Y}\right)^2\right]} \qquad (27)$$

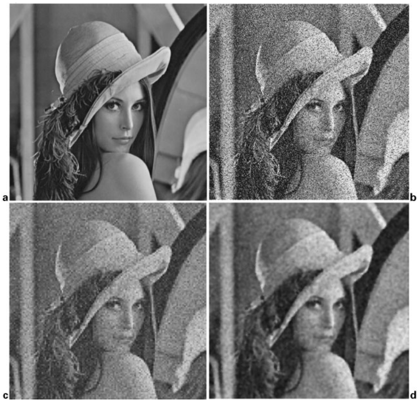
where

$$\begin{aligned} &\overline{X}=\frac{1}{N}\sum_{i=1}^{N}x_i,\ \overline{Y}=\frac{1}{N}\sum_{i=1}^{N}Y_i, \\ &\sigma_x^2=\frac{1}{N-1}\sum_{i=1}^{N}\left(x_i-\overline{X}\right)^2 \\ &\sigma_y^2=\frac{1}{N-1}\sum_{i=1}^{N}\left(y_i-\overline{Y}\right)^2, \\ &\sigma_{xy}=\frac{1}{N-1}\sum_{i=1}^{N}\left(x_i-\overline{X}\right)\left(y_i-\overline{Y}\right) \end{aligned} \qquad (28)$$

The dynamic range of SSIM is $[-1, 1]$. The best value 1 is achieved if and only if $y_i=x_i$ for all $i=1, 2, 3, \ldots, N$. The lowest value of -1 occurs when $y_i=2X-x_i$ for all $i=1, 2, 3, \ldots, N$. This quality index models any distortion as a combination of three different factors: loss of correlation, luminance distortion and contrast distortion. In order to understand this, we rewrite the definition of Q as a product of three components

$$\begin{aligned} &\mathrm{SSIM}=Q=Q_1Q_2Q_3= \\ &\frac{\sigma_{xy}}{\sigma_x\sigma_y}\frac{2\overline{X}\,\overline{Y}}{\left(\overline{X}\right)^2+\left(\overline{Y}\right)^2}\frac{2\sigma_x\sigma_y}{\sigma_x^2+\sigma_y^2}= \\ &s(x,y)^{\alpha}l(x,y)^{\beta}c(x,y)^{\gamma} \end{aligned} \qquad (29)$$

The first component is the correlation coefficient between x and y, which measures the degree of linear



4 Comparison between wavelet thresholding and the proposed method in reducing Salt and Pepper noise: (a) original image; (b) noisy image (Gaussian and Salt & Pepper noise, density: 0.05); (c) wavelet thresholding; (d) proposed method

correlation between x and y, and its dynamic range is $[-1, 1]$. The best value 1 is obtained when $y_i = ax_i + b$ for all $i = 1, 2, 3, \ldots, N$, where a and b are constants and $a > 0$. Even if x and y are linearly related, there still might be relative distortions between them, which are evaluated in the second and third components. The second component, with a value range of [0, 1], measures how much the x and y are close in luminance. It equals 1 if and only if $\overline{X} = \overline{Y}$. σ_x and σ_y can be viewed as an estimate of the contrast of x and y, so the third component measures the similarities between the contrasts of the images. Its range of values is also [0, 1], where the best value 1 is achieved if and only if $\sigma_X = \sigma_Y$.

Parameters α, β and γ are used to adjust the significance of each of the three components. This metric can be implemented for an image as below

$$\mathrm{MSSIM}(x,y) = \frac{1}{M}\sum_{j=1}^{M}\mathrm{SSIM}(x_j,y_j) \tag{30}$$

where X and Y are the reference and the distorted images, respectively, M is the number of local windows in the image, MSSIM is the structural similarity index matrix, and x_j and y_j are the image contents at the jth local window.

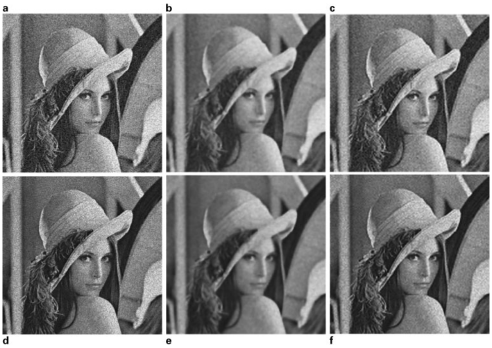
4.3 Peak signal-to-noise ratio

The peak signal-to-noise ratio (PSNR) is a performance metric for the ratio between the maximum possible power of a signal and the power of corrupting noise that affects the fidelity of its representation. It is the most commonly used measure of quality for de-noised signal or image. It is easily defined via the mean square error (MSE) which for two $m \times n$ images I_{Original} and $I_{\text{de-noised}}$ is defined as follows

$$\mathrm{MSE} = \frac{1}{MN}\sum_{i=1}^{N}\sum_{j=1}^{M}\left[I_{\mathrm{original}}(i,j) - I_{\mathrm{denoised}}(i,j)\right]^2 \tag{31}$$

The PSNR is then defined as

$$\mathrm{PSNR} = 10\log_{10}\left\{\frac{[\mathrm{MAX}(B)]^2}{\mathrm{MSE}}\right\} = 20\log_{10}\left[\frac{\mathrm{MAX}(B)}{\mathrm{MSE}}\right] \tag{32}$$



5 Comparison between wavelet thresholding and the proposed method for reducing Gaussian noise (top row), and speckle noise (bottom row): (a) Gaussian noise corrupted (m=0.05, σ=0.02), SSIM=0.2; (b) proposed method, SSIM=0.664; (c) wavelet threshlding, SSIM=0.545; (d) Speckle noise (m=0, σ=0.04), SSIM=0.53; (e) proposed method, SSIM=0.73; (f) wavelet thresholding, SSIM=0.42

Here, MAX(B) is the maximum possible value of a pixel, i.e. for 8-bit pixels MAX(B)=255.

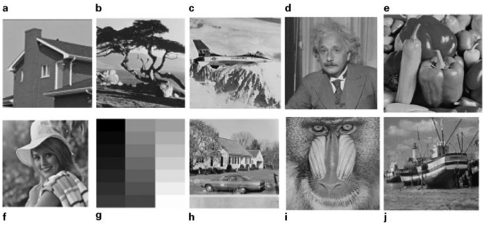
5 SIMULATION RESULTS

To evaluate the performance of the proposed algorithms, they all have been implemented using Matlab, and then applied on a data set of 100 standard images taken from Ref. 28. The evaluation results for one of the images are shown in Fig. 4. In the evaluation process, the image in Fig. 4a was initially affected by a combination of Gaussian and Salt & Pepper noises, and the noisy image is shown in Fig. 4b. This image was initially decomposed using a five-level wavelet transform. The decomposition was performed using Daubechie mother wavelet. This mother wavelet has a good performance in noise reduction, as reported by Zhai and Zhang.[29] The soft thresholding method was applied on the detail coefficients in each level. The complex diffusion equation was applied on the approximate coefficients, as described earlier. Finally, by applying the inverse wavelet transform, the de-noised image was obtained. Results of enhancing Fig. 4b using the two approaches are shown in Fig. 4c and d for comparing their performance visually.

To numerically compare the performance of the proposed method with the existing wavelet thresholding approach, the SSIM criteria is used. SSIM for the de-noised image using the existing wavelet theresholding shown in Fig. 4c is 0.442, and the corresponding value for the de-noised image using the proposed method shown in Fig. 4d is 0.776. Similar experiments have been performed on the same image, corrupted once by Gaussian and then by Speckle noise. The results of these experiments are shown in Fig. 5. These experiments explain the high capability of the proposed method in different kinds of noise reduction.

To further evaluate the performance of the proposed method and the existing methods, several methods were applied on a number of standard images, as shown in Fig. 6. In the evaluation process, the images were corrupted by different types of noise, including Salt & Pepper, Speckle and Gaussian noise.

The performance criteria such as PSNR, FOM and SSIM are applied to comparing the proposed method with several typical and robustness methods such as wavelet thresholding, second order diffusion,



6 Some standard images used for evaluating performance of the proposed method

complex diffusion and NL means. These methods have proven very powerful in image de-noising applications during the recent years. The results of the experiments are shown in Tables 1–3. As can be seen in these tables, the proposed algorithm is functioning better than the other methods. According to FOM, the edges of the images are preserved better. SSIM measure also indicates that the proposed approach has often a better performance in preserving the structure of original images. In some cases, the NL means algorithm has better performance in comparison to the proposed method, but on the other hand, it is a very time-consuming method. The NL means algorithm involves all of the image pixels for de-noising each pixel, and this makes the NL means very slow. Therefore, this algorithm is not efficient in real-time applications. In addition, the method only can reduce additive noise, such as Gaussian noise, but the proposed method is functioning well for additive and also multiplicative noise, such as Speckle.

6 CONCLUSION

Wavelet thresholding is used in many image de-noising techniques. In these techniques, thresholding is not applied on the approximate sub-bands, which have the most crucial information about the signals. Since noise affects both the approximate and detail

Table 1 Quality index (MSSIM), PSNR and FOM criteria for comparing performance of the proposed method with the existing approach in reducing noise from different images (Gaussian noise m=0.2, σ=0.1)

De-noising method	Pic.1	Pic.2	Pic.3	Pic.4	Pic.5	Pic.6	Pic.7	Pic.8	Pic.9	Pic.10
MSSIM values in Gaussian noise reduction										
Noisy image	0.09	0.256	0.21	0.23	0.16	0.08	0.1	0.23	0.15	0.2
Wavelet thresholding	0.26	0.33	0.41	0.32	0.1	0.1	0.45	0.51	0.29	0.35
Non-local mean	0.63	0.47	0.53	0.4	0.51	0.33	0.55	0.34	0.35	0.46
Second order	0.37	0.3	0.35	0.36	0.56	0.26	0.43	0.48	0.37	0.39
Complex diffusion	0.42	0.34	0.31	0.38	0.53	0.31	0.46	0.53	0.39	0.31
Proposed method	**0.6**	**0.52**	**0.52**	**0.45**	**0.58**	**0.35**	**0.51**	**0.61**	**0.45**	**0.41**
PSNR values in Gaussian noise reduction										
Noisy image	11.64	11.35	10.19	12.36	10.11	9.68	10.12	12.65	11.2	12.5
Wavelet thresholding	16.6	18.23	17.42	19.3	17.5	18.1	19.36	21.9	17.8	20.5
Non-local mean	15.1	17.56	17.49	22.32	17.8	18.25	19.2	21.3	18.6	19.62
Second order	15.3	17.12	16.32	21.3	17.6	15.6	18.6	18.9	16.4	18.41
Complex diffusion	15.2	18.16	15.9	22.5	17.5	17.1	18.3	20.6	16.3	18.35
Proposed method	**15.7**	**18.33**	**17.2**	**23.8**	**17.9**	**19.1**	**19.9**	**21.3**	**18.5**	**19.65**
FOM values in Gaussian noise reduction										
Noisy image	0.29	0.36	0.41	0.42	0.32	0.31	0.3	0.42	0.37	0.41
Wavelet thresholding	0.42	0.43	0.68	0.49	0.45	0.42	0.55	0.53	0.53	0.47
Non-local mean	0.51	0.63	0.81	0.57	0.57	0.45	0.62	0.58	0.67	0.57
Second order	0.51	0.52	0.71	0.51	0.51	0.46	0.53	0.57	0.57	0.46
Complex diffusion	0.53	0.61	0.72	0.53	0.53	0.45	0.51	0.56	0.63	0.52
Proposed method	**0.56**	**0.65**	**0.78**	**0.59**	**0.63**	**0.49**	**0.61**	**0.63**	**0.68**	**0.56**

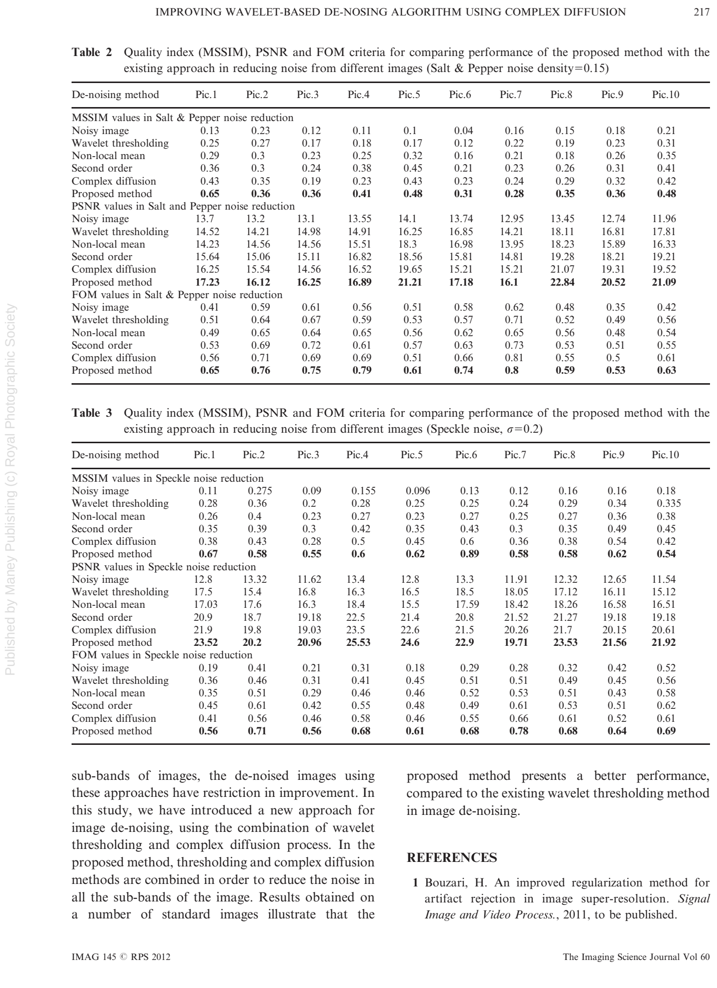


Table 2 Quality index (MSSIM), PSNR and FOM criteria for comparing performance of the proposed method with the existing approach in reducing noise from different images (Salt & Pepper noise density=0.15)

De-noising method	Pic.1	Pic.2	Pic.3	Pic.4	Pic.5	Pic.6	Pic.7	Pic.8	Pic.9	Pic.10
MSSIM values in Salt & Pepper noise reduction										
Noisy image	0.13	0.23	0.12	0.11	0.1	0.04	0.16	0.15	0.18	0.21
Wavelet thresholding	0.25	0.27	0.17	0.18	0.17	0.12	0.22	0.19	0.23	0.31
Non-local mean	0.29	0.3	0.23	0.25	0.32	0.16	0.21	0.18	0.26	0.35
Second order	0.36	0.3	0.24	0.38	0.45	0.21	0.23	0.26	0.31	0.41
Complex diffusion	0.43	0.35	0.19	0.23	0.43	0.23	0.24	0.29	0.32	0.42
Proposed method	**0.65**	**0.36**	**0.36**	**0.41**	**0.48**	**0.31**	**0.28**	**0.35**	**0.36**	**0.48**
PSNR values in Salt and Pepper noise reduction										
Noisy image	13.7	13.2	13.1	13.55	14.1	13.74	12.95	13.45	12.74	11.96
Wavelet thresholding	14.52	14.21	14.98	14.91	16.25	16.85	14.21	18.11	16.81	17.81
Non-local mean	14.23	14.56	14.56	15.51	18.3	16.98	13.95	18.23	15.89	16.33
Second order	15.64	15.06	15.11	16.82	18.56	15.81	14.81	19.28	18.21	19.21
Complex diffusion	16.25	15.54	14.56	16.52	19.65	15.21	15.21	21.07	19.31	19.52
Proposed method	**17.23**	**16.12**	**16.25**	**16.89**	**21.21**	**17.18**	**16.1**	**22.84**	**20.52**	**21.09**
FOM values in Salt & Pepper noise reduction										
Noisy image	0.41	0.59	0.61	0.56	0.51	0.58	0.62	0.48	0.35	0.42
Wavelet thresholding	0.51	0.64	0.67	0.59	0.53	0.57	0.71	0.52	0.49	0.56
Non-local mean	0.49	0.65	0.64	0.65	0.56	0.62	0.65	0.56	0.48	0.54
Second order	0.53	0.69	0.72	0.61	0.57	0.63	0.73	0.53	0.51	0.55
Complex diffusion	0.56	0.71	0.69	0.69	0.51	0.66	0.81	0.55	0.5	0.61
Proposed method	**0.65**	**0.76**	**0.75**	**0.79**	**0.61**	**0.74**	**0.8**	**0.59**	**0.53**	**0.63**

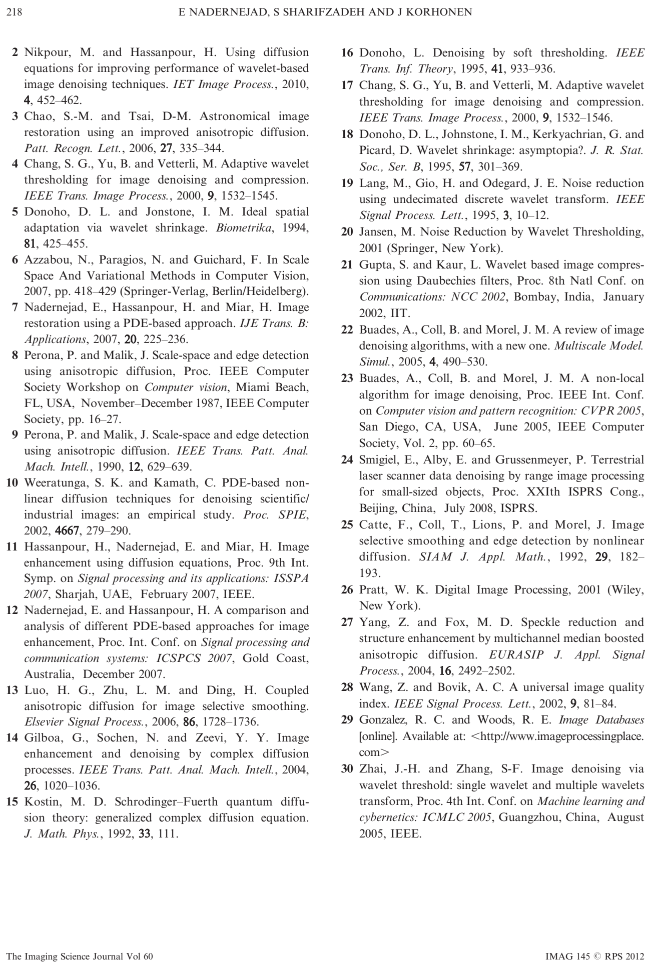
Table 3 Quality index (MSSIM), PSNR and FOM criteria for comparing performance of the proposed method with the existing approach in reducing noise from different images (Speckle noise, σ=0.2)

De-noising method	Pic.1	Pic.2	Pic.3	Pic.4	Pic.5	Pic.6	Pic.7	Pic.8	Pic.9	Pic.10
MSSIM values in Speckle noise reduction										
Noisy image	0.11	0.275	0.09	0.155	0.096	0.13	0.12	0.16	0.16	0.18
Wavelet thresholding	0.28	0.36	0.2	0.28	0.25	0.25	0.24	0.29	0.34	0.335
Non-local mean	0.26	0.4	0.23	0.27	0.23	0.27	0.25	0.27	0.36	0.38
Second order	0.35	0.39	0.3	0.42	0.35	0.43	0.3	0.35	0.49	0.45
Complex diffusion	0.38	0.43	0.28	0.5	0.45	0.6	0.36	0.38	0.54	0.42
Proposed method	**0.67**	**0.58**	**0.55**	**0.6**	**0.62**	**0.89**	**0.58**	**0.58**	**0.62**	**0.54**
PSNR values in Speckle noise reduction										
Noisy image	12.8	13.32	11.62	13.4	12.8	13.3	11.91	12.32	12.65	11.54
Wavelet thresholding	17.5	15.4	16.8	16.3	16.5	18.5	18.05	17.12	16.11	15.12
Non-local mean	17.03	17.6	16.3	18.4	15.5	17.59	18.42	18.26	16.58	16.51
Second order	20.9	18.7	19.18	22.5	21.4	20.8	21.52	21.27	19.18	19.18
Complex diffusion	21.9	19.8	19.03	23.5	22.6	21.5	20.26	21.7	20.15	20.61
Proposed method	**23.52**	**20.2**	**20.96**	**25.53**	**24.6**	**22.9**	**19.71**	**23.53**	**21.56**	**21.92**
FOM values in Speckle noise reduction										
Noisy image	0.19	0.41	0.21	0.31	0.18	0.29	0.28	0.32	0.42	0.52
Wavelet thresholding	0.36	0.46	0.31	0.41	0.45	0.51	0.51	0.49	0.45	0.56
Non-local mean	0.35	0.51	0.29	0.46	0.46	0.52	0.53	0.51	0.43	0.58
Second order	0.45	0.61	0.42	0.55	0.48	0.49	0.61	0.53	0.51	0.62
Complex diffusion	0.41	0.56	0.46	0.58	0.46	0.55	0.66	0.61	0.52	0.61
Proposed method	**0.56**	**0.71**	**0.56**	**0.68**	**0.61**	**0.68**	**0.78**	**0.68**	**0.64**	**0.69**

sub-bands of images, the de-noised images using these approaches have restriction in improvement. In this study, we have introduced a new approach for image de-noising, using the combination of wavelet thresholding and complex diffusion process. In the proposed method, thresholding and complex diffusion methods are combined in order to reduce the noise in all the sub-bands of the image. Results obtained on a number of standard images illustrate that the proposed method presents a better performance, compared to the existing wavelet thresholding method in image de-noising.

REFERENCES

1 Bouzari, H. An improved regularization method for artifact rejection in image super-resolution. *Signal Image and Video Process.*, 2011, to be published.



2 Nikpour, M. and Hassanpour, H. Using diffusion equations for improving performance of wavelet-based image denoising techniques. *IET Image Process.*, 2010, **4**, 452–462.

3 Chao, S.-M. and Tsai, D-M. Astronomical image restoration using an improved anisotropic diffusion. *Patt. Recogn. Lett.*, 2006, **27**, 335–344.

4 Chang, S. G., Yu, B. and Vetterli, M. Adaptive wavelet thresholding for image denoising and compression. *IEEE Trans. Image Process.*, 2000, **9**, 1532–1545.

5 Donoho, D. L. and Jonstone, I. M. Ideal spatial adaptation via wavelet shrinkage. *Biometrika*, 1994, **81**, 425–455.

6 Azzabou, N., Paragios, N. and Guichard, F. In Scale Space And Variational Methods in Computer Vision, 2007, pp. 418–429 (Springer-Verlag, Berlin/Heidelberg).

7 Nadernejad, E., Hassanpour, H. and Miar, H. Image restoration using a PDE-based approach. *IJE Trans. B: Applications*, 2007, **20**, 225–236.

8 Perona, P. and Malik, J. Scale-space and edge detection using anisotropic diffusion, Proc. IEEE Computer Society Workshop on *Computer vision*, Miami Beach, FL, USA, November–December 1987, IEEE Computer Society, pp. 16–27.

9 Perona, P. and Malik, J. Scale-space and edge detection using anisotropic diffusion. *IEEE Trans. Patt. Anal. Mach. Intell.*, 1990, **12**, 629–639.

10 Weeratunga, S. K. and Kamath, C. PDE-based non-linear diffusion techniques for denoising scientific/industrial images: an empirical study. *Proc. SPIE*, 2002, **4667**, 279–290.

11 Hassanpour, H., Nadernejad, E. and Miar, H. Image enhancement using diffusion equations, Proc. 9th Int. Symp. on *Signal processing and its applications: ISSPA 2007*, Sharjah, UAE, February 2007, IEEE.

12 Nadernejad, E. and Hassanpour, H. A comparison and analysis of different PDE-based approaches for image enhancement, Proc. Int. Conf. on *Signal processing and communication systems: ICSPCS 2007*, Gold Coast, Australia, December 2007.

13 Luo, H. G., Zhu, L. M. and Ding, H. Coupled anisotropic diffusion for image selective smoothing. *Elsevier Signal Process.*, 2006, **86**, 1728–1736.

14 Gilboa, G., Sochen, N. and Zeevi, Y. Y. Image enhancement and denoising by complex diffusion processes. *IEEE Trans. Patt. Anal. Mach. Intell.*, 2004, **26**, 1020–1036.

15 Kostin, M. D. Schrodinger–Fuerth quantum diffusion theory: generalized complex diffusion equation. *J. Math. Phys.*, 1992, **33**, 111.

16 Donoho, L. Denoising by soft thresholding. *IEEE Trans. Inf. Theory*, 1995, **41**, 933–936.

17 Chang, S. G., Yu, B. and Vetterli, M. Adaptive wavelet thresholding for image denoising and compression. *IEEE Trans. Image Process.*, 2000, **9**, 1532–1546.

18 Donoho, D. L., Johnstone, I. M., Kerkyachrian, G. and Picard, D. Wavelet shrinkage: asymptopia?. *J. R. Stat. Soc., Ser. B*, 1995, **57**, 301–369.

19 Lang, M., Gio, H. and Odegard, J. E. Noise reduction using undecimated discrete wavelet transform. *IEEE Signal Process. Lett.*, 1995, **3**, 10–12.

20 Jansen, M. Noise Reduction by Wavelet Thresholding, 2001 (Springer, New York).

21 Gupta, S. and Kaur, L. Wavelet based image compression using Daubechies filters, Proc. 8th Natl Conf. on *Communications: NCC 2002*, Bombay, India, January 2002, IIT.

22 Buades, A., Coll, B. and Morel, J. M. A review of image denoising algorithms, with a new one. *Multiscale Model. Simul.*, 2005, **4**, 490–530.

23 Buades, A., Coll, B. and Morel, J. M. A non-local algorithm for image denoising, Proc. IEEE Int. Conf. on *Computer vision and pattern recognition: CVPR 2005*, San Diego, CA, USA, June 2005, IEEE Computer Society, Vol. 2, pp. 60–65.

24 Smigiel, E., Alby, E. and Grussenmeyer, P. Terrestrial laser scanner data denoising by range image processing for small-sized objects, Proc. XXIth ISPRS Cong., Beijing, China, July 2008, ISPRS.

25 Catte, F., Coll, T., Lions, P. and Morel, J. Image selective smoothing and edge detection by nonlinear diffusion. *SIAM J. Appl. Math.*, 1992, **29**, 182–193.

26 Pratt, W. K. Digital Image Processing, 2001 (Wiley, New York).

27 Yang, Z. and Fox, M. D. Speckle reduction and structure enhancement by multichannel median boosted anisotropic diffusion. *EURASIP J. Appl. Signal Process.*, 2004, **16**, 2492–2502.

28 Wang, Z. and Bovik, A. C. A universal image quality index. *IEEE Signal Process. Lett.*, 2002, **9**, 81–84.

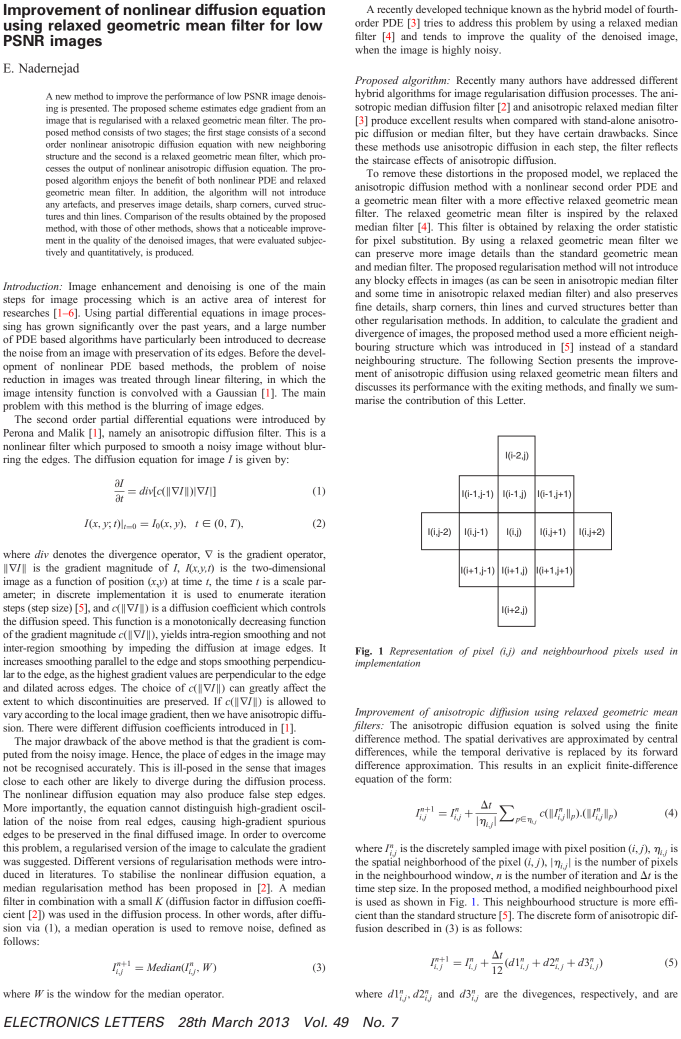
29 Gonzalez, R. C. and Woods, R. E. *Image Databases* [online]. Available at: <http://www.imageprocessingplace.com>

30 Zhai, J.-H. and Zhang, S-F. Image denoising via wavelet threshold: single wavelet and multiple wavelets transform, Proc. 4th Int. Conf. on *Machine learning and cybernetics: ICMLC 2005*, Guangzhou, China, August 2005, IEEE.

A.9 An Improvement of Nonlinear Diffusion Equation using Relaxed Geometric Mean Filter for Low PSNR Images

E. Nadernejad, “Improvement of Nonlinear Diffusion Equation Using Relaxed Geometric Mean Filter for Low PSNR Images”, *Electronics Letters*, vol. 49, no. 7, pp. 457–458, 2013

Reference: [9]



Improvement of nonlinear diffusion equation using relaxed geometric mean filter for low PSNR images

E. Nadernejad

A new method to improve the performance of low PSNR image denoising is presented. The proposed scheme estimates edge gradient from an image that is regularised with a relaxed geometric mean filter. The proposed method consists of two stages; the first stage consists of a second order nonlinear anisotropic diffusion equation with new neighboring structure and the second is a relaxed geometric mean filter, which processes the output of nonlinear anisotropic diffusion equation. The proposed algorithm enjoys the benefit of both nonlinear PDE and relaxed geometric mean filter. In addition, the algorithm will not introduce any artefacts, and preserves image details, sharp corners, curved structures and thin lines. Comparison of the results obtained by the proposed method, with those of other methods, shows that a noticeable improvement in the quality of the denoised images, that were evaluated subjectively and quantitatively, is produced.

Introduction: Image enhancement and denoising is one of the main steps for image processing which is an active area of interest for researches [1–6]. Using partial differential equations in image processing has grown significantly over the past years, and a large number of PDE based algorithms have particularly been introduced to decrease the noise from an image with preservation of its edges. Before the development of nonlinear PDE based methods, the problem of noise reduction in images was treated through linear filtering, in which the image intensity function is convolved with a Gaussian [1]. The main problem with this method is the blurring of image edges.

The second order partial differential equations were introduced by Perona and Malik [1], namely an anisotropic diffusion filter. This is a nonlinear filter which purposed to smooth a noisy image without blurring the edges. The diffusion equation for image I is given by:

$$\frac{\partial I}{\partial t} = div[c(\|\nabla I\|)|\nabla I|] \tag{1}$$

$$I(x, y; t)|_{t=0} = I_0(x, y), \quad t \in (0, T), \tag{2}$$

where *div* denotes the divergence operator, ∇ 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, the time t is a scale parameter; in discrete implementation it is used to enumerate iteration steps (step size) [5], and $c(\|\nabla I\|)$ is a diffusion coefficient which controls the diffusion speed. This function is a monotonically decreasing function of the gradient magnitude $c(\|\nabla I\|)$, yields intra-region smoothing and not inter-region smoothing 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. There were different diffusion coefficients introduced in [1].

The major drawback of the above method is that the gradient is computed from the noisy image. Hence, the place of edges in the image may not be recognised accurately. This is ill-posed in the sense that images close to each other are likely to diverge during the diffusion process. The nonlinear diffusion equation may also produce false step edges. More importantly, the equation cannot distinguish high-gradient oscillation of the noise from real edges, causing high-gradient spurious edges to be preserved in the final diffused image. In order to overcome this problem, a regularised version of the image to calculate the gradient was suggested. Different versions of regularisation methods were introduced in literatures. To stabilise the nonlinear diffusion equation, a median regularisation method has been proposed in [2]. A median filter in combination with a small K (diffusion factor in diffusion coefficient [2]) was used in the diffusion process. In other words, after diffusion via (1), a median operation is used to remove noise, defined as follows:

$$I_{i,j}^{n+1} = Median(I_{i,j}^{n}, W) \tag{3}$$

where W is the window for the median operator.

A recently developed technique known as the hybrid model of fourth-order PDE [3] tries to address this problem by using a relaxed median filter [4] and tends to improve the quality of the denoised image, when the image is highly noisy.

Proposed algorithm: Recently many authors have addressed different hybrid algorithms for image regularisation diffusion processes. The anisotropic median diffusion filter [2] and anisotropic relaxed median filter [3] produce excellent results when compared with stand-alone anisotropic diffusion or median filter, but they have certain drawbacks. Since these methods use anisotropic diffusion in each step, the filter reflects the staircase effects of anisotropic diffusion.

To remove these distortions in the proposed model, we replaced the anisotropic diffusion method with a nonlinear second order PDE and a geometric mean filter with a more effective relaxed geometric mean filter. The relaxed geometric mean filter is inspired by the relaxed median filter [4]. This filter is obtained by relaxing the order statistic for pixel substitution. By using a relaxed geometric mean filter we can preserve more image details than the standard geometric mean and median filter. The proposed regularisation method will not introduce any blocky effects in images (as can be seen in anisotropic median filter and some time in anisotropic relaxed median filter) and also preserves fine details, sharp corners, thin lines and curved structures better than other regularisation methods. In addition, to calculate the gradient and divergence of images, the proposed method used a more efficient neighbouring structure which was introduced in [5] instead of a standard neighbouring structure. The following Section presents the improvement of anisotropic diffusion using relaxed geometric mean filters and discusses its performance with the exiting methods, and finally we summarise the contribution of this Letter.

		I(i-2,j)		
	I(i-1,j-1)	I(i-1,j)	I(i-1,j+1)	
I(i,j-2)	I(i,j-1)	I(i,j)	I(i,j+1)	I(i,j+2)
	I(i+1,j-1)	I(i+1,j)	I(i+1,j+1)	
		I(i+2,j)		

Fig. 1 *Representation of pixel (i,j) and neighbourhood pixels used in implementation*

Improvement of anisotropic diffusion using relaxed geometric mean filters: The anisotropic diffusion equation is solved using the finite difference method. 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:

$$I_{i,j}^{n+1} = I_{i,j}^{n} + \frac{\Delta t}{|\eta_{i,j}|}\sum\nolimits_{p\in\eta_{i,j}} c(\|I_{i,j}^{n}\|_p).(\|I_{i,j}^{n}\|_p) \tag{4}$$

where $I_{i,j}^{n}$ is the discretely sampled image with pixel position (i, j), $\eta_{i,j}$ is the spatial neighborhood of the pixel (i, j), $|\eta_{i,j}|$ is the number of pixels in the neighbourhood window, n is the number of iteration and Δt is the time step size. In the proposed method, a modified neighbourhood pixel is used as shown in Fig. 1. This neighbourhood structure is more efficient than the standard structure [5]. The discrete form of anisotropic diffusion described in (3) is as follows:

$$I_{i,j}^{n+1} = I_{i,j}^{n} + \frac{\Delta t}{12}(d1_{i,j}^{n} + d2_{i,j}^{n} + d3_{i,j}^{n}) \tag{5}$$

where $d1_{i,j}^{n}, d2_{i,j}^{n}$ and $d3_{i,j}^{n}$ are the divegences, respectively, and are calcluted as follows:

$$d1^{n}_{i,j} = c^{n}_{i+1,j}(I^{n}_{i+1,j} - I^{n}_{i,j}) + c^{n}_{i-1,j}(I^{n}_{i-1,j} - I^{n}_{i,j}) \\ + c^{n}_{i,j+1}(I^{n}_{i,j+1} - I^{n}_{i,j}) + c^{n}_{i,j-1}(I^{n}_{i,j-1} - I^{n}_{i,j}) \tag{6}$$

The other divegence values are calculated in the same way. The relaxed geometric mean filter [4] can be used in combination with (5) to remove noise. The proposed hybrid method is defined as follows:

$$I^{n+1}_{i,j} = RGM_{\alpha\omega}\left(I^{n}_{i,j} + \frac{\Delta t}{12}(d1^{n}_{i,j} + d2^{n}_{i,j} + d3^{n}_{i,j})\right) \tag{7}$$

where RGM is the relaxed geometric mean filter with lower bound α and upper bound ω. The working of a relaxed geometric filter is as follows: it has two bounds, a lower bound (α) and an upper bound (ω), defines a sub list inside the window $[W(i,j)]_{(.)}$, which contains the grey levels that we assume to be good enough not to be filtered. If $Y(i,j)$ is the output of a relaxed median filter, then $Y(i,j)$ can be written as:

$$\begin{aligned} Y_{(i,j)} &= RGM_{\alpha,\omega}\{W_{(i,j)}\} \\ &= \begin{cases} X_{(i,j)} & \mathit{if}\ X_{(i,j)} \in [[W_{(i,j)}]_{\alpha}[W_{(i,j)}]_{\omega}] \\ [W_{(i,j)}]_{m} & \text{otherwise} \end{cases} \end{aligned} \tag{8}$$

$$W_{(i,j)} = \left[\prod_{(s,t)\in\mathbb{Z}^{m}} I(s,t)\right]^{\frac{1}{mn}} \tag{9}$$

where $[W_{(i,j)}]_{m}$ is the median value of the samples inside the window $W_{(i,j)}$. The sliding window $W_{(i,j)}$ is $W_{(i,j)} = \{X_{(i+r,j+r)} : r \in W\}$. To be the window located at position i. The lower bound and upper bounds for the relaxed median used in the experiments are 3 and 5, respectively.

When anisotropic diffusion is applied to the images, the areas having small gradients are smoothed, and those having large gradients (edges and noise, if any) remain undiffused and the blocky effects can be avoided. The gradients generated by noise can be subsequently removed by a relaxed geometric mean filter without affecting the image structure. However if the gradients are generated by edges, the geometric mean filter will not affect them. So as iteration continues, the nonlinear diffusion equation removes the low level noise and subsequently the relaxed median filter removes the impulsive noise spikes. The proposed method preserves image structure much better than the other similar methods.

Fig. 2 shows a comparative analysis of the proposed method with the existing anisotropic median diffusion filters. The noise distributions in different regions are also different. It can be seen that the methods in [2] and [3] remove noise but generate false edges and smooth the edge regions. Fig. 2*d* shows that this artefact can be eliminated by using the proposed method.

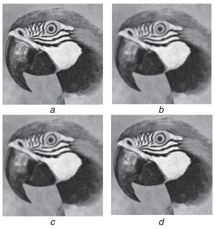


Fig. 2 *Simulation results on real image after five iteration*

a original image
b algorithm by [2]
c algorithm by [3]
d proposed algorithm

Conclusion: A method for noise removal based on the anisotropic diffusion equation has been proposed. The artefacts generated by the anisotropic diffusion method can be can be considerably reduced by this approach. The method preserves curve-like structures and edges much better than the existing methods. The filter has been tested against standard test images (by adding different levels of noise). The analysis shows that the anisotropic diffusion method can be improved by applying the proposed approach.

12 November 2012
doi: 10.1049/el.2012.3946

E. Nadernejad (*Department of Photonics Engineering, Coding and Visual Communication, Technical University of Denmark, Ørsted Plads building 343, Copenhagen, 2800, Denmark*)

E-mail: ehsan.nadernejad@ieee.org

References

1 Perona, P., and Malik, J.: 'Scale-space and edge detection using anisotropic diffusion', *IEEE Trans. Pattern Anal. Mach. Intel.*, 1990, **12**, (7), pp. 629–639

2 Ling, J., and Bovik, A.C.: 'Smoothing low-SNR molecular images via anisotropic median diffusion', *IEEE Trans. Med. Imaging*, 2002, **21**, (4), pp. 377–384

3 Rajan, J., Kannan, K., and Kaimal, M.R.: 'An improved hybrid for molecular image denoising', *J Math Imaging Vis.,* 2008, **31**, (1), pp. 73–79

4 Hamza, A.B., and Krim, H.: 'Image denoising: a nonlinear robust statistical approach', *IEEE Trans. Signal Process*, 2001, **49**, (12), pp. 3045–3054

5 Nadernejad, E., Hassanpour, H., and Miar, H.: 'Image restoration using a PDE-based approach,', *Int. J. Eng, Trans B: Apps*, 2007, **20**, (3), pp. 225–236

6 Portilla, J., Strela, V., Wainwright, M.J., and Simoncelli, E.P.: 'Image denoising using scale mixtures of Gaussians in the wavelet domain', *IEEE Trans. Image Process*, 2003, **12**, (11), pp. 1338–1351

A.10 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*, vol. 9, no. 8, pp. 656–665, 2013

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*, Søren Forchhammer, *Member, IEEE*, and Xiaolin Wu, *Fellow, IEEE*

***Abstract*—In this paper, we present a framework for modeling color liquid crystal displays (LCDs) having local light-emitting diode (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, light-emitting diode (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

Manuscript received December 21, 2012; revised February 11, 2013; accepted March 13, 2013. Date of publication April 05, 2013; date of current version August 07, 2013.

N. Burini, E. Nadernejad, J. Korhonen, and S. Forchhammer are with DTU Fotonik, Technical University of Denmark, Kongens Lyngby, 2800, Denmark (e-mail: nibu@fotonik.dtu.dk; ehsa@fotonik.dtu.dk; jark@fotonik.dtu.dk; sofo@fotonik.dtu.dk).

X. Wu is with the Department of Electrical and Computer Engineering, McMaster University, Hamilton, ON L8S4L8, Canada (e-mail: xwu@ece.mcmaster.ca).

Color versions of one or more of the figures are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JDT.2013.2253544

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 in to 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]

$$l = bt. \qquad (1)$$

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 ε, 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

$$l = bt + \varepsilon b(1 - t) \qquad (2)$$

or alternatively

$$l = bt_o \qquad (3)$$

where t_o is the observed transmittance, as opposed to ideal transmittance t; t_o can be expressed as

$$t_o = (1 - \varepsilon)t + \varepsilon. \qquad (4)$$

The model, (2)–(4), allows each pixel to have a different ε value, as leakage increases with the viewing angle θ. 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 ε 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

$$b = \sum\nolimits_{k=1}^{N} r_k h_k \qquad (5)$$

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, (5) for all pixels is

$$\boldsymbol{b} = \boldsymbol{Hr} \qquad (6)$$

where the column vector $\boldsymbol{b}$ has a b value per pixel, the *influence matrix* $\boldsymbol{H}$ (N columns and a row per pixel) represents the PSFs and $\boldsymbol{r}$ 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 (1) by replacing l with the target pixel luminance l_y

$$t_C = \frac{l_y}{b}. \qquad (7)$$

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 ε due to leakage. Equation (7) should thus be rewritten as

$$t_C = \begin{cases} 1, & \text{if } l_y/b > 1 \text{ (Cond.I)} \\ 0, & \text{if } l_y/b < \varepsilon \text{ (Cond.II)} \\ (l_y/b - \varepsilon)/(1 - \varepsilon), & \text{otherwise.} \end{cases} \qquad (8)$$

Condition I defines the occurrence of *clipping*, when the LCs cannot fully compensate the backlight reduction creating

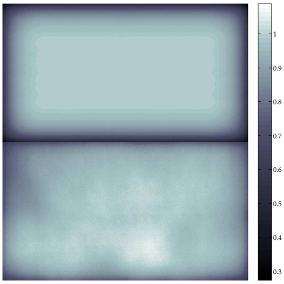


Fig. 1. Example of the modeled (top) and the measured backlight (bottom).

clipped pixels that appear dimmer than intended. Condition II defines the occurrence of *leakage*, when compensated pixels are brighter then intended. The observed physical transmittance can be computed from (4), by replacing t with t_C.

It should be noted that (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 nonlinearly: the sensitivity to luminance variations decreases as the luminance increases. In general, a perceptual response can be expressed as

$$l_U = g(l) \tag{9}$$

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 (1)–(8). In this work, we shall approximate g with the Gamma function, i.e.,

$$l_U = l^{\frac{1}{\gamma}}. \tag{10}$$

The inverse response is simply given by

$$l = l_U^{\gamma}. \tag{11}$$

In this work, for simplicity, we approximate the HVS response to luminance with (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^2 [12]; for higher values, larger γ 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^2 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 $\boldsymbol{y}$ and on the actual output $\boldsymbol{x}$ rendered on the locally backlit LCD. The elements of $\boldsymbol{x}$ are given by (3); the backlight $\boldsymbol{b}$ is given by (6), the transmittances $\boldsymbol{t}$ are limited between ε 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 $\boldsymbol{x}$-$\boldsymbol{y}$, has been extended to include an error weighting vector $\boldsymbol{w}$, 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:

$$p = \frac{\sum_{k=1}^{N} r_k}{N}. \tag{12}$$

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 $\boldsymbol{w}$ 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 $f(\boldsymbol{y}, \boldsymbol{x}, \boldsymbol{w}, q, p)$ 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 $\boldsymbol{y}$-$\boldsymbol{x}$ in ℓ_1. The following sections present three formulations of backlight optimization problems where the cost includes the norm of $\boldsymbol{y}$-$\boldsymbol{x}$ 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 Section III-B–III-D.

B. Single Component

The following formulation considers the case where $\boldsymbol{y}$ and $\boldsymbol{x}$ have one color component, i.e., grayscale images or the luma component of a color frame

$$\begin{aligned} \min \quad & \|(\boldsymbol{y} - \boldsymbol{x}) \circ \boldsymbol{w}\| + q \times p \\ \text{s.t.} \quad & \boldsymbol{x} = \boldsymbol{b} \circ \boldsymbol{t} \\ & \boldsymbol{b} = \boldsymbol{Hr} \\ & \varepsilon \le t_k \le 1, \quad k = 1, \ldots, mn \\ & 0 \le r_k \le 1, \quad k = 1, \ldots, N \end{aligned} \tag{13}$$

where the $\circ$ operator defines element-wise multiplication, and the first two conditions derive from (3) and (6), respectively. After optimization, the r_k values determine the optimal backlight. The problem may be formulated in linear form [3]:

$$\begin{aligned} \min \quad & \|\boldsymbol{\lambda}\| + q \times p \\ \text{s.t.} \quad & \boldsymbol{b} = \boldsymbol{Hr} \\ & \boldsymbol{\lambda} \ge ((\boldsymbol{\varepsilon} \circ \boldsymbol{b}) - \boldsymbol{y}) \circ \boldsymbol{w} \\ & \boldsymbol{\lambda} \ge (\boldsymbol{y} - \boldsymbol{b}) \circ \boldsymbol{w} \\ & \boldsymbol{\lambda} \ge 0 \\ & 0 \le r_k \le 1, \quad k = 1, \ldots, N. \end{aligned} \tag{14}$$

If q is set to 0 and the elements of $\boldsymbol{w}$ to 1, the problem is equivalent to [3]. If additionally the elements of $\boldsymbol{\varepsilon}$ 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 $\boldsymbol{x}$ and $\boldsymbol{y}$ 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

$$\begin{aligned} \min \quad & \left\| \sum_{i=R,G,B} (\boldsymbol{y}_i - \boldsymbol{x}_i) \circ \boldsymbol{w}_i \right\| + q \times p \\ \text{s.t.} \quad & \boldsymbol{x}_i = \boldsymbol{b} \circ \boldsymbol{t}_i; \quad i = R, G, B \\ & \boldsymbol{b} = \boldsymbol{Hr} \\ & \varepsilon \le t_{ki} \le 1; k = 1, \ldots, mn; i = R, G, B \\ & 0 \le r_k \le 1; \quad k = 1, \ldots, N. \end{aligned} \tag{15}$$

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 (13) specifies an upper and a lower threshold for the variables $\boldsymbol{\lambda}$ 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 $\boldsymbol{y}$ with $\boldsymbol{y_{min}}$ or $\boldsymbol{y_{max}}$; $\boldsymbol{y_{min}}$ is the vector of the minimum values of each RGB triplet, while $\boldsymbol{y_{max}}$ is the vector of the maximum values:

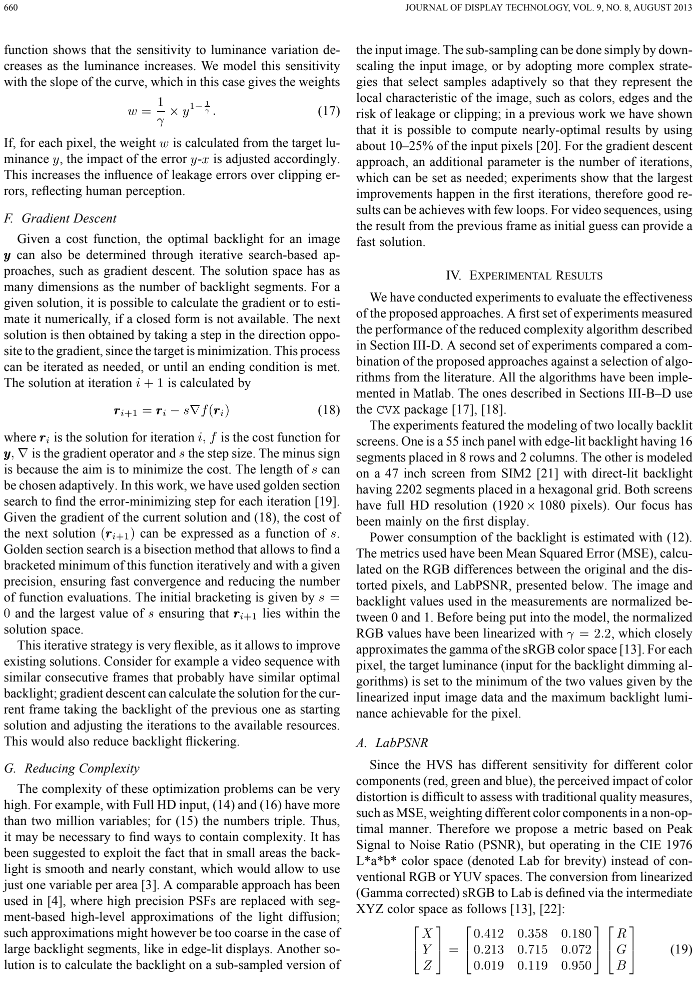
$$\begin{aligned} \min \quad & \|\boldsymbol{\lambda}\| + q \times p \\ \text{s.t.} \quad & \boldsymbol{b} = \boldsymbol{Hr} \\ & \boldsymbol{\lambda} \ge ((\boldsymbol{\varepsilon} \circ \boldsymbol{b}) - \boldsymbol{y_{min}}) \circ \boldsymbol{w_{min}} \\ & \boldsymbol{\lambda} \ge (\boldsymbol{y_{max}} - \boldsymbol{b}) \circ \boldsymbol{w_{max}} \\ & \boldsymbol{\lambda} \ge 0 \\ & 0 \le r_k \le 1, \quad k = 1, \ldots, N. \end{aligned} \tag{16}$$

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 $\boldsymbol{\lambda}$ 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 $\boldsymbol{y}$ and $\boldsymbol{x}$ 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 $\boldsymbol{w}$ introduced in the extended formulation of (13)–(16).

As mentioned in Section II-D, in this work we approximate the HVS response to luminance using (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

$$w = \frac{1}{\gamma} \times y^{1-\frac{1}{\gamma}}. \tag{17}$$

If, for each pixel, the weight w is calculated from the target luminance y, the impact of the error y-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 $\boldsymbol{y}$ 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

$$\boldsymbol{r}_{i+1} = \boldsymbol{r}_i - s\nabla f(\boldsymbol{r}_i) \tag{18}$$

where $\boldsymbol{r}_i$ is the solution for iteration i, f is the cost function for $\boldsymbol{y}$, ∇ 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 (18), the cost of the next solution $(\boldsymbol{r}_{i+1})$ 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 $\boldsymbol{r}_{i+1}$ 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, (14) and (16) have more than two million variables; for (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 achieves 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 backlit 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 (1920 × 1080 pixels). Our focus has been mainly on the first display.

Power consumption of the backlight is estimated with (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]:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.412 & 0.358 & 0.180 \\ 0.213 & 0.715 & 0.072 \\ 0.019 & 0.119 & 0.950 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{19}$$

$$\begin{aligned} L* &= 116 \cdot f(Y/Y_n) - 16 \\ a* &= 500 \cdot [f(X/X_n) - f(Y/Y_n)] \\ b* &= 200 \cdot [f(Y/Y_n) - f(Z/Z_n)] \end{aligned} \tag{20}$$

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 ΔE, that is the perceived difference between two colors, considering both luminance and chrominance differences [22]

$$\Delta E = \sqrt{\Delta L*^2 + \Delta a*^2 + \Delta b*^2} \tag{21}$$

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 ΔE:

$$\text{LabPSNR} = 10 \cdot \log_{10} \left(\frac{(\Delta E_{\max})^2}{\frac{1}{mn} \sum_{i=1}^{n} \sum_{j=1}^{m} \Delta E(i,j)^2} \right) \tag{22}$$

where $\Delta E(i,j)$ is the color difference at pixel position (i,j), given by (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 (15) by setting ε 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.

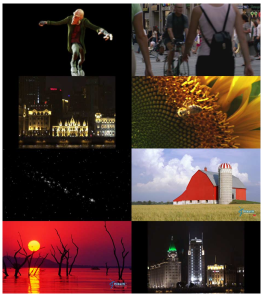


Fig. 2. Sample of images used for the experiments: (left to right, top to bottom) *Man*, *Pedestrian*, *City1*, *Sunflower*, *Stars*, *Barn*, *Sunset* and *City2*.

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 (16) compared with the full RGB problem (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 ε valued 0.0002 and 0.001 and for both ℓ_1 and ℓ_2. The power penalty q was set to zero. The input images have been downscaled 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 I shows the results for downscaling factor 4 (edge-lit screen) and 10 (direct-lit screen). In ℓ_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 ℓ_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

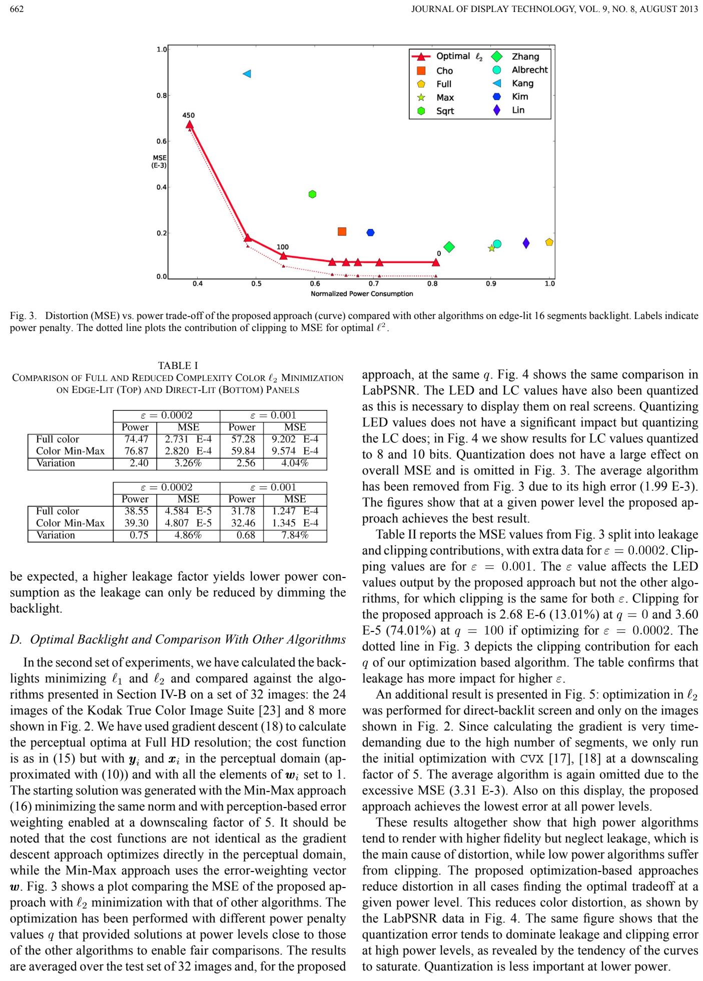


Fig. 3. Distortion (MSE) vs. power trade-off of the proposed approach (curve) compared with other algorithms on edge-lit 16 segments backlight. Labels indicate power penalty. The dotted line plots the contribution of clipping to MSE for optimal ℓ^2.

TABLE I
COMPARISON OF FULL AND REDUCED COMPLEXITY COLOR ℓ_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	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%

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 ℓ_1 and ℓ_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 (18) to calculate the perceptual optima at Full HD resolution; the cost function is as in (15) but with $\boldsymbol{y}_i$ and $\boldsymbol{x}_i$ in the perceptual domain (approximated with (10)) and with all the elements of $\boldsymbol{w}_i$ set to 1. The starting solution was generated with the Min-Max approach (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 $\boldsymbol{w}$. Fig. 3 shows a plot comparing the MSE of the proposed approach with ℓ_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. Fig. 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 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 $\varepsilon = 0.0002$. Clipping values are for $\varepsilon = 0.001$. The ε value affects the LED values output by the proposed approach but not the other algorithms, for which clipping is the same for both ε. 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 Fig. 3 depicts the clipping contribution for each q of our optimization based algorithm. The table confirms that leakage has more impact for higher ε.

An additional result is presented in Fig. 5: optimization in ℓ_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 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.

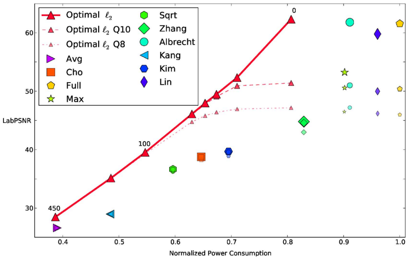


Fig. 4. Quality (LabPSNR) vs. power trade-off of the proposed approach (curve) compared with other algorithms on edge-lit 16 segments backlight. Labels indicate power penalty. The smaller markers indicate quantized results for 10 bits (middle size) and 8 bits (smallest); power levels and symbols are the same as the non-quantized result.

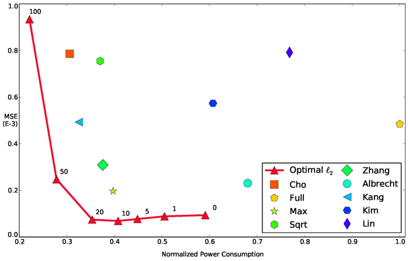


Fig. 5. Distortion (MSE) vs. power trade-off of the proposed approach (curve) compared with other algorithms on direct-lit 2202 segments backlight (downscaled 5 times). Labels indicate power penalty.

TABLE II
CONTRIBUTION OF LEAKAGE AND CLIPPING TO MSE (EDGE-LIT)

	$\varepsilon = 0.0002$		$\varepsilon = 0.001$		
	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

E. Subjective Experiments

Many studies have proven that simple distortion measures, such as MSE and PSNR, do not always predict the perceived 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

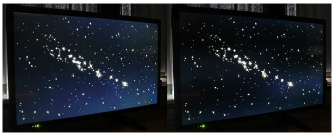


Fig. 6. Displayed results of the Stars image (as in Table III) for backlight algorithm *Albrecht* (left) and gradient descent (*GDq0*, right).

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)		
GDq0	45.17	0.948
Albrecht	45.78	0.985
Zhang	38.01	0.932
GDq200	33.52	0.763
Kang	29.32	0.752
GDq400	23.86	0.578
Cho	22.76	0.635

Parrot (SROCC 0.96)		
GDq0	51.42	0.813
Albrecht	51.74	0.840
Zhang	50.00	0.811
GDq100	44.92	0.648
Cho	38.50	0.660
GDq600	25.37	0.440
Kang	25.02	0.447

Volcano (SROCC 0.83)		
Albrecht	38.98	0.660
GDq0	38.19	0.513
Zhang	38.04	0.767
Cho	25.48	0.439
Kang	22.62	0.388
GDq400	26.39	0.324

Lizard (SROCC 0.89)		
GDq1	46.46	0.870
Albrecht	47.35	0.981
Zhang	36.28	0.904
GDq400	37.06	0.644
Cho	25.35	0.684
Kang	24.57	0.658

Beach (SROCC 0.89)		
GDq1	51.91	0.730
Albrecht	52.92	0.855
Cho	39.11	0.767
Zhang	45.25	0.800
GDq800	23.17	0.483
Kang	20.42	0.448

Diver (SROCC 0.70)		
GDq0	46.90	0.702
GDq400	30.12	0.546
Albrecht	44.11	0.825
Zhang	32.54	0.740
Cho	23.66	0.560

Stars (SROCC 0.40)		
GDq0	39.29	0.304
Albrecht	36.89	0.982
Kang	38.58	0.504
Cho	38.34	0.499

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 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. Fig. 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 nonlinear 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. 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 Trans. Electron.*, 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, pp. 166–173, 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, Feb. 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, Jun. 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. Aydın, R. Mantiuk, and H.-P. Seidel, "Extending quality metrics to full luminance range images," in *Proc. SPIE*, 2008, vol. 6806, pp. 6806–6810.

[13] "sRGB color space," [Online]. Available: http://www.w3.org/Graphics/Color/sRGB/

[14] C. Poynton, *A Technical Introduction to Digital Video*. New York, NY, USA: Wiley, 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*, 2011, vol. 7866, 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,” ver. 1.21, Apr. 2011 [Online]. Available: http://cvx.com/cvx

[18] M. Grant and S. Boyd, , V. Blondel, S. Boyd, and H. Kimura, Eds., “Graph implementations for nonsmooth convex programs,” in *Recent Advances in Learning and Control*, ser. Lecture Notes in Control Inf. Sci.. : Springer-Verlag, 2008, pp. 95–110.

[19] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C*, 2nd ed. New York, NY, USA: Cambridge Univ. Press, 1992.

[20] N. Burini, E. Nadernejad, J. Korhonen, S. Forchhammer, and X. Wu, “Speedup of optimization-based approach to local backlight dimming of HDR displays,” in *Proc. SPIE*, 2012, vol. 8436, pp. 84 360B–1.

[21] “SIM2 HDR TV,” [Online]. Available: http://www.sim2.com/HDR/

[22] J.-R. Ohm, *Multimedia Communication Technology: Representation, Transmission and Identification of Multimedia Signals*. New York, NY, USA: Springer, 2004.

[23] “Kodak true color image suite,” [Online]. Available: http://r0k.us/graphics/kodak/

[24] “The consumer digital video library,” [Online]. Available: 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*, 2013, vol. 8652, p. 86520V.

Nino Burini (M’10) received the B.S. and M.S. degrees in computer science and engineering from the University of Ferrara, Italy, in 2006 and 2009, respectively, and is currently pursuing the Ph.D. degree at DTU Fotonik, the Department of Photonics Engineering of the Technical University of Denmark, Kongens Lyngby, Denmark.

His research interests include LCD backlight, display technology, image and video coding and processing.

Ehsan Nadernejad (M’12) received the B.S. and M.S. degrees in electronic engineering from Mazandaran University, Iran, in 2003 and 2007, respectively, and is currently working toward the Ph.D. degree at DTU Fotonik, Department of Photonics Engineering of the Technical University of Denmark, Kongens Lyngby, Denmark.

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 the M.S. (Eng.) degree in information engineering from University of Oulu, Finland, in 2001, and the 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, Lyngby, 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.

Søren Forchhammer (M’04) received the M.S. degree in engineering and the Ph.D. degree from the Technical University of Denmark, Lyngby, Denmark, 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.

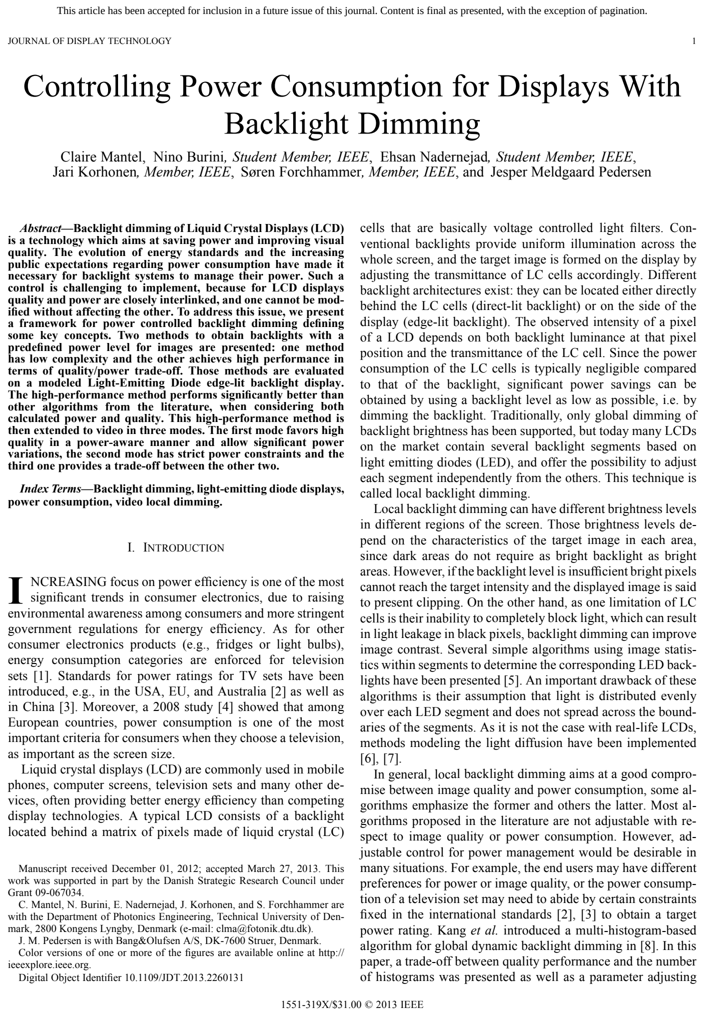
He started his academic career in 1988, and he has been with the faculty of the University of Western Ontario, Canada, and of New York Polytechnic University, Brooklyn, NY, USA. He is currently Professor at the Department of Electrical and Computer Engineering, McMaster University, Hamilton, ON, Canada, 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 an associate editor of the IEEE TRANSACTIONS ON IMAGE PROCESSING.

A.11 Controlling Power Consumption for Displays With Backlight Dimming

C. Mantel, N. Burini, **E. Nadernejad**, J. Korhonen, S. Forchhammer, and J. Pedersen, "Controlling Power Consumption for Displays With Backlight Dimming", *Display Technology, Journal of*, vol. 9, no. 12, pp. 933–941, 2013

Reference: [11]



Controlling Power Consumption for Displays With Backlight Dimming

Claire Mantel, Nino Burini, *Student Member, IEEE*, Ehsan Nadernejad, *Student Member, IEEE*, Jari Korhonen, *Member, IEEE*, Søren Forchhammer, *Member, IEEE*, and Jesper Meldgaard Pedersen

***Abstract*—Backlight dimming of Liquid Crystal Displays (LCD) is a technology which aims at saving power and improving visual quality. The evolution of energy standards and the increasing public expectations regarding power consumption have made it necessary for backlight systems to manage their power. Such a control is challenging to implement, because for LCD displays quality and power are closely interlinked, and one cannot be modified without affecting the other. To address this issue, we present a framework for power controlled backlight dimming defining some key concepts. Two methods to obtain backlights with a predefined power level for images are presented: one method has low complexity and the other achieves high performance in terms of quality/power trade-off. Those methods are evaluated on a modeled Light-Emitting Diode edge-lit backlight display. The high-performance method performs significantly better than other algorithms from the literature, when considering both calculated power and quality. This high-performance method is then extended to video in three modes. The first mode favors high quality in a power-aware manner and allow significant power variations, the second mode has strict power constraints and the third one provides a trade-off between the other two.**

***Index Terms*—Backlight dimming, light-emitting diode displays, power consumption, video local dimming.**

I. Introduction

INCREASING focus on power efficiency is one of the most significant trends in consumer electronics, due to raising environmental awareness among consumers and more stringent government regulations for energy efficiency. As for other consumer electronics products (e.g., fridges or light bulbs), energy consumption categories are enforced for television sets [1]. Standards for power ratings for TV sets have been introduced, e.g., in the USA, EU, and Australia [2] as well as in China [3]. Moreover, a 2008 study [4] showed that among European countries, power consumption is one of the most important criteria for consumers when they choose a television, as important as the screen size.

Liquid crystal displays (LCD) are commonly used in mobile phones, computer screens, television sets and many other devices, often providing better energy efficiency than competing display technologies. A typical LCD consists of a backlight located behind a matrix of pixels made of liquid crystal (LC) cells that are basically voltage controlled light filters. Conventional backlights provide uniform illumination across the whole screen, and the target image is formed on the display by adjusting the transmittance of LC cells accordingly. Different backlight architectures exist: they can be located either directly behind the LC cells (direct-lit backlight) or on the side of the display (edge-lit backlight). The observed intensity of a pixel of a LCD depends on both backlight luminance at that pixel position and the transmittance of the LC cell. Since the power consumption of the LC cells is typically negligible compared to that of the backlight, significant power savings can be obtained by using a backlight level as low as possible, i.e. by dimming the backlight. Traditionally, only global dimming of backlight brightness has been supported, but today many LCDs on the market contain several backlight segments based on light emitting diodes (LED), and offer the possibility to adjust each segment independently from the others. This technique is called local backlight dimming.

Local backlight dimming can have different brightness levels in different regions of the screen. Those brightness levels depend on the characteristics of the target image in each area, since dark areas do not require as bright backlight as bright areas. However, if the backlight level is insufficient bright pixels cannot reach the target intensity and the displayed image is said to present clipping. On the other hand, as one limitation of LC cells is their inability to completely block light, which can result in light leakage in black pixels, backlight dimming can improve image contrast. Several simple algorithms using image statistics within segments to determine the corresponding LED backlights have been presented [5]. An important drawback of these algorithms is their assumption that light is distributed evenly over each LED segment and does not spread across the boundaries of the segments. As it is not the case with real-life LCDs, methods modeling the light diffusion have been implemented [6], [7].

In general, local backlight dimming aims at a good compromise between image quality and power consumption, some algorithms emphasize the former and others the latter. Most algorithms proposed in the literature are not adjustable with respect to image quality or power consumption. However, adjustable control for power management would be desirable in many situations. For example, the end users may have different preferences for power or image quality, or the power consumption of a television set may need to abide by certain constraints fixed in the international standards [2], [3] to obtain a target power rating. Kang *et al.* introduced a multi-histogram-based algorithm for global dynamic backlight dimming in [8]. In this paper, a trade-off between quality performance and the number of histograms was presented as well as a parameter adjusting

Manuscript received December 01, 2012; accepted March 27, 2013. This work was supported in part by the Danish Strategic Research Council under Grant 09-067034.

C. Mantel, N. Burini, E. Nadernejad, J. Korhonen, and S. Forchhammer are with the Department of Photonics Engineering, Technical University of Denmark, 2800 Kongens Lyngby, Denmark (e-mail: clma@fotonik.dtu.dk).

J. M. Pedersen is with Bang&Olufsen A/S, DK-7600 Struer, Denmark.

Color versions of one or more of the figures are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JDT.2013.2260131

the quality level, and as a result the power level. The backlight dimming algorithm presented in [9] includes the effect of leakage and computes the best possible backlight (in terms of MSE) for any chosen trade-off between quality and power. To the best of our knowledge, this is the only published method supplying a direct mean to trade-off power and quality. However, this method does not provide an explicit mechanism to specify a target power level. Indeed, the resulting power is determined by interactions between the chosen power/quality trade-off and the input image characteristics.

This paper extends the work on optimization based backlight dimming [9] to videos and especially introduces a framework for a higher level monitoring and management of power and quality. Our focus is here to design and evaluate methods to enforce power consumption constraints while at the same time maintaining the best possible quality. In this paper, we present two modes for computing the backlight of an image at a given power consumption: the first mode aims at low complexity and an acceptable quality whereas the other mode achieves the best possible power-quality performance at the price of a higher complexity. Different types of power management, corresponding to different power constraints, are also implemented for the high-performance mode in the context of video sequences. We use measurements from a 55$''$ edge-lit display to simulate and evaluate the performance of backlight dimming methods.

The rest of this paper is organized as follows. Firstly, Section II presents selected works on backlight dimming models and algorithms. Then a general framework to address power control and management of backlight dimming is introduced in Section III. Section IV describes and evaluates the two proposed methods of power-control for backlight dimming on images: one is a high-performance method and the other a low-complexity method. Finally, in Section V an adaptation of the high-performance power control to video sequences is presented for the basic setting as well as for both long-term and short-term constraints on the power consumption.

II. Backlight Dimming

In backlight architectures, the light from each backlight segment is diffused and mixed with the light coming from the other segments on the diffuser plate, located behind the LC cells. The light from an individual light source follows a point spreading function (PSF, also called light spreading function), that depends on the optical properties of the diffuser plate.

A. Backlight Display Modeling

Backlight intensity, b, can be modeled by multiplying normalized PSF and the backlight level r, and then summing the contributions from all backlight segments [10]

$$b(i) = \sum_{k=1}^{N} r_k \cdot h_k(i) \tag{1}$$

where N is the number of segments, r_k is the intensity of segment k and $h_k(i)$ is the value of the PSF at the image position indexed by i. The observed luminance level $y(i)$ at the image position indexed by i can be expressed as a function of the LCs transmittance $t(i)$ and the backlight level $b(i)$

$$y(i) = b(i) \cdot t(i) \tag{2}$$

where both transmittance and backlight have been normalized to $[0, 1]$. For color images, (2) can be applied to red, green and blue (RGB) components separately.

B. Backlight Dimming Algorithms

The different approaches presented for backlight dimming can be classified in two general categories: some algorithms use statistic features of the target image [5], [11] while others additionally use the two-dimensional PSF of the light source (LED) on the diffuser plate [6]. The first category includes some very simple algorithms such as the average, the square root and the maximum methods, for which the intensity of each backlight segment is set respectively to the average, the normalized square-root of the average and the maximum of pixel values in the considered segment [5]. The two main weaknesses of the Max algorithm are that it is very sensitive to noise as it depends on the value of only one pixel, and that it does not consider dark pixels, i.e. it does not take leakage into account. Using the average pixel value as backlight intensity causes in turn heavy clipping to bright pixels in dark segments, such as stars in an image of a night sky. Better trade-off between leakage and clipping is usually achieved by the Square-root algorithm instead of direct average, but this method is also vulnerable to the actual pixel values distribution.

Albrecht *et al.* [6] presented a backlight dimming algorithm that minimizes the energy consumption under the constraint of being clipper-free. They use PSF of light sources to model the light diffusion. First, 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 pixel requiring the largest increase in luminance to be rendered properly is found, and the intensity of the most influential LED for this pixel is increased. If this LED is already at its maximum, then the second most influential LED is used, and so on until all pixels are sufficiently lit. The final 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.

Kang *et al.* [8] introduced a global backlight dimming algorithm based on multiple histograms. This algorithm allows to select a target PSNR, which in turn constrains the power consumption. A target MSE is calculated using the inverse of the target PSNR. Then the backlight luminance is calculated to achieve the best amount of clipping defined by matching the total squared errors of each histogram and maximum total square error of each block with the target MSE iteratively.

C. Optimization Based Backlight Dimming

In [12], we have presented a model of the image rendered by a LCD screen with local backlight dimming that accounts for both leakage and clipping defects. This model allows to precisely evaluate the difference between the original image and the displayed one. Thanks to this model, backlight dimming can be

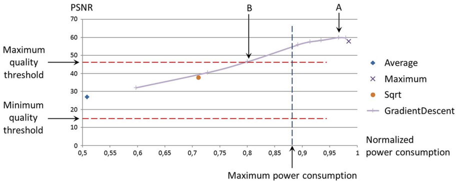


Fig. 1. Power and quality thresholds in relation to quality as a function of power for image k08 of the Kodak database.

approached as an optimization issue with the objective of minimizing the error between an original image and the displayed one [7]. The mean square error (MSE) can be computed as

$$\frac{1}{N}\sum_{i}(y_i - x_i)^2 \tag{3}$$

where x_i is the target value and y_i the rendered value at image position i, obtained by combining the backlight from (1), with the LC transmittance from (2). In (3), and throughout the paper, we map the relation of perceived luminance and physical luminance using a gamma of 2.2.

Modeling backlight dimming as an optimization problem and minimizing (3) is possible but complex [7]. Valid approximations can be downscaling the input image or considering only the color component that contribute the most to the error [13]. An iterative gradient-descent optimization algorithm was also presented in [10], where tests showed that computing the optimal solution on a downsampled version of an image and then applying the gradient descent on the full scale version allows to reach the optimal solution.

Furthermore, the possibility to account for the power consumption was introduced in [9] by adding it to the error that is minimized. The resulting cost function, C is expressed by

$$C = \sum_{i}(y_i - x_i)^2 + q \cdot p \tag{4}$$

where q is a power weight, putting a penalty on the power value, and the power p is computed as the average LED value over all backlight segments

$$p = \frac{1}{N} \cdot \sum_{k=1}^{N} r_k. \tag{5}$$

Measures on a 55 inch LCD edge-lit display showed the validity of using the average LED values as a power consumption estimate for the backlight, as the rest of the TV set has an approximately constant power consumption.

III. Framework for a Power-Constrained Backlight Dimming Algorithm

As can be derived from (1) to (3), the quality of a displayed image depends both on its content and its backlight. It is therefore not possible to separate the power consumption of a backlight from the quality of the rendered image and any constraints on one of them influences the other. Plots of quality (e.g., expressed in terms of MSE or PSNR) as a function of power consumption are a graphical representation of this dependency. For bright images, quality improves as the power is increased since there is no possible leakage. For dark images, this is true only up to a certain power value above which the increase in leakage will dominate over the decrease in clipping in the overall quality. In this section, we present a general framework and define four concepts to address control of power consumption. The first two notions concern energy: maximum power consumption and power buffer. The last two concepts deal with quality: the maximum and minimum quality thresholds. Fig. 1 illustrates those concepts in relation to quality and power performance.

A. Virtual Power Buffer

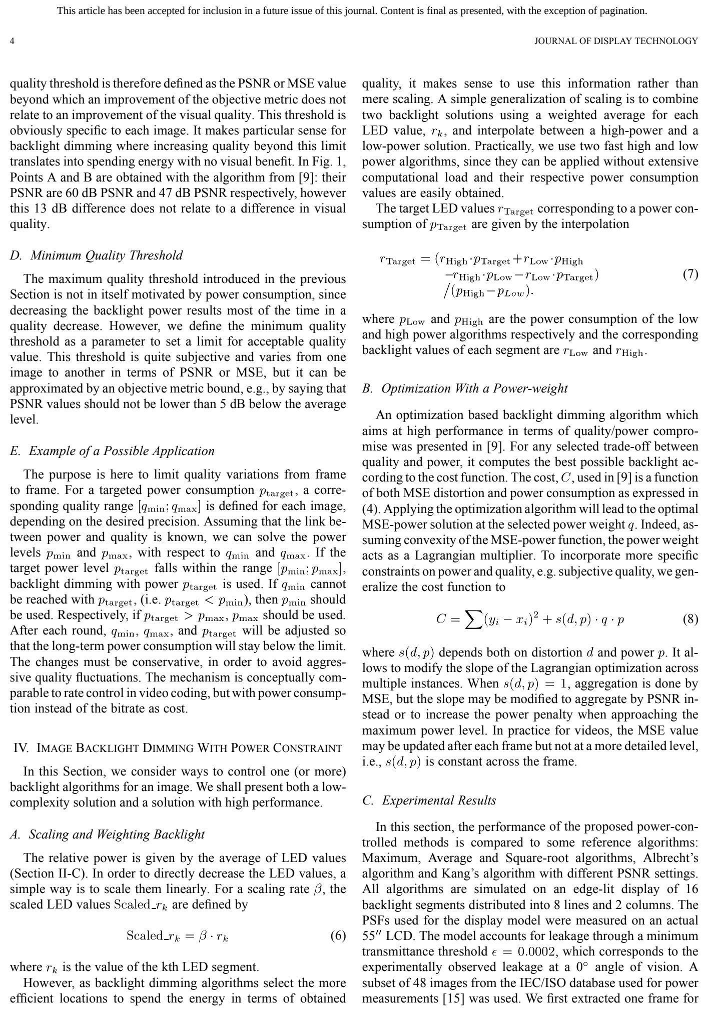
By analogy with the virtual buffer concept used in video coding rate control (e.g., the Decoded Picture Buffer in H.264 standard [14]), we define a virtual power buffer that keeps track of the energy consumed by the backlight of a display. At any moment, the power buffer contains the accumulated power of all the previously displayed frames, which may be compared to a "target" power consumption with a latency due to the buffer. This corresponds to the way energy categories are assigned to TV sets, i.e. comparing the total power consumption over a few minutes of display to different thresholds [15]. Another motivation for a virtual power buffer is to monitor the temperature of a TV set to avoid overheating.

B. Maximum Power Consumption

In addition to controlling overall power consumption, it may also be useful for TV manufacturers to have a limit (lower than that of full backlight) on the instantaneous energy consumed. Indeed, it can serve to find the best dimensioning of the electronic components or to estimate the maximal heat diffused by the LEDs. Therefore, we define maximum power consumption (MPC) as the threshold of maximum authorized power. The virtual power buffer and the maximum power may be combined to avoid overheating, e.g., under warm conditions for a display operating in direct sunlight.

C. Maximum Quality Threshold

Generally, an improvement of PSNR implies a corresponding improvement of the perceptual quality. However, at some point quality saturates and an increase in PSNR has no effect on the visual quality (same for small MSE values). The concept of high



quality threshold is therefore defined as the PSNR or MSE value beyond which an improvement of the objective metric does not relate to an improvement of the visual quality. This threshold is obviously specific to each image. It makes particular sense for backlight dimming where increasing quality beyond this limit translates into spending energy with no visual benefit. In Fig. 1, Points A and B are obtained with the algorithm from [9]: their PSNR are 60 dB PSNR and 47 dB PSNR respectively, however this 13 dB difference does not relate to a difference in visual quality.

D. Minimum Quality Threshold

The maximum quality threshold introduced in the previous Section is not in itself motivated by power consumption, since decreasing the backlight power results most of the time in a quality decrease. However, we define the minimum quality threshold as a parameter to set a limit for acceptable quality value. This threshold is quite subjective and varies from one image to another in terms of PSNR or MSE, but it can be approximated by an objective metric bound, e.g., by saying that PSNR values should not be lower than 5 dB below the average level.

E. Example of a Possible Application

The purpose is here to limit quality variations from frame to frame. For a targeted power consumption p_{target}, a corresponding quality range $[q_{\min}; q_{\max}]$ is defined for each image, depending on the desired precision. Assuming that the link between power and quality is known, we can solve the power levels $p_{\min}$ and $p_{\max}$, with respect to $q_{\min}$ and $q_{\max}$. If the target power level p_{target} falls within the range $[p_{\min}; p_{\max}]$, backlight dimming with power p_{target} is used. If $q_{\min}$ cannot be reached with p_{target}, (i.e. $p_{\text{target}} < p_{\min}$), then $p_{\min}$ should be used. Respectively, if $p_{\text{target}} > p_{\max}$, $p_{\max}$ should be used. After each round, $q_{\min}$, $q_{\max}$, and p_{target} will be adjusted so that the long-term power consumption will stay below the limit. The changes must be conservative, in order to avoid aggressive quality fluctuations. The mechanism is conceptually comparable to rate control in video coding, but with power consumption instead of the bitrate as cost.

IV. Image Backlight Dimming With Power Constraint

In this Section, we consider ways to control one (or more) backlight algorithms for an image. We shall present both a low-complexity solution and a solution with high performance.

A. Scaling and Weighting Backlight

The relative power is given by the average of LED values (Section II-C). In order to directly decrease the LED values, a simple way is to scale them linearly. For a scaling rate β, the scaled LED values $\text{Scaled\_}r_k$ are defined by

$$\text{Scaled\_}r_k = \beta \cdot r_k \tag{6}$$

where r_k is the value of the kth LED segment.

However, as backlight dimming algorithms select the more efficient locations to spend the energy in terms of obtained quality, it makes sense to use this information rather than mere scaling. A simple generalization of scaling is to combine two backlight solutions using a weighted average for each LED value, r_k, and interpolate between a high-power and a low-power solution. Practically, we use two fast high and low power algorithms, since they can be applied without extensive computational load and their respective power consumption values are easily obtained.

The target LED values r_{Target} corresponding to a power consumption of p_{Target} are given by the interpolation

$$\begin{aligned} r_{\text{Target}} = (&r_{\text{High}} \cdot p_{\text{Target}} + r_{\text{Low}} \cdot p_{\text{High}} \\ &- r_{\text{High}} \cdot p_{\text{Low}} - r_{\text{Low}} \cdot p_{\text{Target}}) \\ &/ (p_{\text{High}} - p_{Low}). \end{aligned} \tag{7}$$

where p_{Low} and p_{High} are the power consumption of the low and high power algorithms respectively and the corresponding backlight values of each segment are r_{Low} and r_{High}.

B. Optimization With a Power-weight

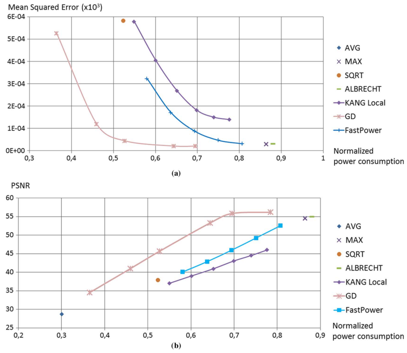
An optimization based backlight dimming algorithm which aims at high performance in terms of quality/power compromise was presented in [9]. For any selected trade-off between quality and power, it computes the best possible backlight according to the cost function. The cost, C, used in [9] is a function of both MSE distortion and power consumption as expressed in (4). Applying the optimization algorithm will lead to the optimal MSE-power solution at the selected power weight q. Indeed, assuming convexity of the MSE-power function, the power weight acts as a Lagrangian multiplier. To incorporate more specific constraints on power and quality, e.g. subjective quality, we generalize the cost function to

$$C = \sum (y_i - x_i)^2 + s(d,p) \cdot q \cdot p \tag{8}$$

where $s(d,p)$ depends both on distortion d and power p. It allows to modify the slope of the Lagrangian optimization across multiple instances. When $s(d,p) = 1$, aggregation is done by MSE, but the slope may be modified to aggregate by PSNR instead or to increase the power penalty when approaching the maximum power level. In practice for videos, the MSE value may be updated after each frame but not at a more detailed level, i.e., $s(d,p)$ is constant across the frame.

C. Experimental Results

In this section, the performance of the proposed power-controlled methods is compared to some reference algorithms: Maximum, Average and Square-root algorithms, Albrecht's algorithm and Kang's algorithm with different PSNR settings. All algorithms are simulated on an edge-lit display of 16 backlight segments distributed into 8 lines and 2 columns. The PSFs used for the display model were measured on an actual 55″ LCD. The model accounts for leakage through a minimum transmittance threshold $\epsilon = 0.0002$, which corresponds to the experimentally observed leakage at a 0° angle of vision. A subset of 48 images from the IEC/ISO database used for power measurements [15] was used. We first extracted one frame for



(a)

(b)

Fig. 2. Quality as a function of power for a 48 images subset of the IEC database for literature algorithms and the proposed power-constrained algorithms. In (a), the quality is measured with MSE applied without quantization while in (b) PSNR with 10-bits quantization is used.

each sequence of the database and then selected 48 images that represent every type of content present in the database.[1]

The first two steps of Albrecht's algorithm [16] (out of three, the third one being optional) have been implemented, which produces clipper-free results. Kang's algorithm [8] is a global dimming algorithm, but we have generalized it to local dimming. This local version, called '*local Kang*,' was applied with 6 PSNR targets ranging from 30 dB to 55 dB with one value every 5 dB and each backlight segment divided into four blocks to compute the backlight luminance. To implement the weighting of LED values described in Section IV-A, called '*fast-power*' in the figures, we chose the maximum algorithm as the high-power algorithm and the square-root as the low-power.

The results are displayed through Quality(power) curves averaged over all images for two quality measures: MSE computed on unquantized variables and PSNR for 10 bits quantized LC values are shown in Fig. 2(a) and 2(b), respectively.

V. Controlling the Backlight for Video Sequences

A. Constant Quality/Power Tradeoff Mode

This mode consists in applying the gradient descent algorithm at a constant power weight q. It globally follows the same process as the one described in Section IV-B. The only notable difference is that, except for the first frame of a sequence, the LED values obtained for a frame are used as the starting values for the next one.

When it is required to enforce a maximum power consumption (MPC), an obvious solution is to couple this constant power weight mode with a scaling of LED values when they are above the MPC. However, scaling the LED moves us away from the optimal solution achieved with the gradient descent algorithm. The further the obtained power is from the target one, the higher the decrease in quality is.

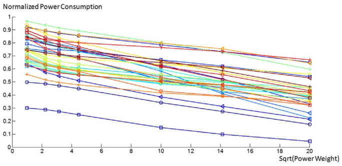


Fig. 3. Obtained power consumption as a function of the square root of power weight for the Kodak database images.

B. Controlling the Output Power

As discussed in the previous Section, directly scaling the LEDs is not a fully satisfying solution to efficiently constrain the power consumption. This observation led to considering another way of modifying energy consumption. As explained in Section IV-B, the higher the power weight q is, the lower the output power. However, the link between power weight and the power actually reached is not straightforward but depends highly on the image content and the power weight is related to the slope of the MSE-power curve. Therefore, this Section is a study of how to modify best the power weight in order to achieve the target power consumption.

The curves shown in Fig. 3 represent the power achieved by our dimming algorithm as a function of the power weight for

[1] The chosen subset, IECSubset.png, is shown in the supplemental material available online at http://ieeexplore.ieee.org

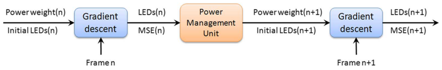


Fig. 4. General scheme of our power management unit.

the images of the Kodak database [17]. Observing a linear relation between the power, p, and $\sqrt{q}$, it seemed possible to model the relationship between the power weight fed to the gradient descent algorithm and the obtained power using the following formula:

$$q = (m \cdot p + k)^2 \tag{9}$$

where q is the power weight and p the power. This formula contains two degrees of freedom: m and k that can be evaluated through linear regression, provided that at least two different (q, p) points are available. We evaluated this model on the Kodak database [17] using 7 different power weights (1, 5, 10, 20, 100, 200 and 400) and obtained an average correlation of the fit of 0.9887, i.e., very close to linear.

C. Short-Term Power Constraint Mode

The method described here has a short reaction time, meaning that it checks the power consumed by the backlight for each image and reacts without any delay if it is above the MPC. The reaction is twofold: firstly the LED values are linearly scaled to the MPC, secondly the power weight q given to our algorithm is increased in order to decrease the obtained power for the next frame (see Section V-B).

In order to modify the power weight q, (9) is used. Practically, if the power consumption obtained for the nth frame of a sequence is higher than the MPC, then for the $(n+1)$th frame the new power weight q_{n+1} is defined as

$$q_{n+1} = (m \cdot MPC - p_n) + \sqrt{q_n})^2 . \tag{10}$$

The ideal model is stable over time while fitting to the power consumption trend. In practice, consecutive images can vary significantly and increasing the power weight from frame n to frame $n+1$ can nevertheless result in an increase of power consumption $p_{n+1} > p_n$, inverting the sign of model slope m. Another risk is that having multiple observations for a specific power weight q, may lead to a slope m, which is artificially close to 0. To prevent those issues only one power value for each power weight is used in the regression. After each frame, the power estimate $p_{q,n}$ corresponding to the last used power weight q is updated by

$$p_{q,n+1} = \frac{(p_{q,n} + p_{n+1})}{2}. \tag{11}$$

The slope of the model is also constrained by $m < -10$, which corresponds to the smallest slope obtained when testing this model on the Kodak database.

D. Long-Term Power Constraint Mode

The general scheme of our control is presented in Fig. 4. For each frame, the gradient descent needs initial LED values and a power weight q as inputs. The first frame is initialized with the optimal LED values computed on a downscaled version of the first frame [13]. The following frames use the LED of the previous frame for initialization. One of many standard backlight algorithms could also have been used.

After the backlight of a frame is computed, the m and k parameters of the $q(p)$ model expressed in (10) are updated through a linear regression between p and $\sqrt{q}$ values. For reasons explained in Section V-C, a single power value is associated to each previously exercised power weight. As this method works on a longer period of time, the average of the power consumptions obtained with a power weight are associated with it. The difference $Diff$ between the obtained and ideal power consumptions is computed as

$$Diff = \sum_{k \in [1,N]} p_k - n \cdot p_{\mathrm{Target}} \tag{12}$$

where N is the frame number. It is compared to the buffer size for the current frame $N \times T_B$, where T_B is the buffer threshold of the virtual power buffer presented in Section III-A. The power management unit can then adopt 3 different behaviors.

- If $\|Diff\| < N \cdot T_B$, i.e. if we are inside the desired bounds, the new power weight is computed by evaluating (9) with the target power.
- If $Diff > N \cdot T_B$, the power weight is updated by

$$q = \left(m \cdot (p_{\mathrm{target}} - T_B) + k\right)^2 . \tag{13}$$

- Finally, if $Diff < -N \cdot T_B$, quality may be increased and the new power weight is

$$q = \left(m \cdot (p_{\mathrm{target}} + T_B/2) + k\right)^2 . \tag{14}$$

The buffer threshold is used to decrease the target power because it represents a significant power variation. As our main concern is power-savings, over-consumption of power is more tightly controlled and fought than under-consumption.

E. Experimental Results

1) Experiment description: To assess the performance of our methods on videos, we tested them on the first 100 frames of 7 Full-HD (1920×1080) sequences on the LCD model described in Section IV-C. Two of those sequences are from the IEC power measurement database [15] (*LiftOff* and *Fireworks*) and the remaining five come from the Consumer Digital Video Library [18] (*BbScore*, *BBBunny*, *Anemone*, *Diver*, *NightLights*). Those sequences present varying characteristics in terms of luminance,

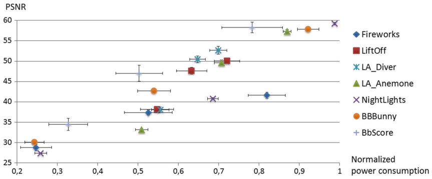


Fig. 5. PSNR as a function of power for each sequence for method *CstPW* at 3 different power weights (1, 100, and 400).

Methods	Measures		BbScore			BigBuckBunny			Fireworks			Anemone			Diver			LiftOff			NightLights			Average		
			p1	p100	p400	p1	p100	p400	p1	p100	P400	p1	p100	p400	p1	p100	p400	p1	p100	p400	p1	p100	p400	p1	p100	p400
CstPW	PSNR	Avg	58,24	47,03	34,48	57,85	42,69	30,11	41,63	37,40	28,83	57,33	49,59	33,21	52,65	50,46	38,09	50,03	47,64	38,18	59,22	40,80	27,44	53,85	45,09	32,91
		Min	55,91	43,41	31,99	57,40	40,92	28,86	40,28	36,21	27,08	56,64	48,41	32,85	50,59	48,78	36,58	48,91	46,36	36,97	58,98	40,32	27,07	52,67	43,49	31,63
	Power	Avg	0,783	0,503	0,327	0,922	0,540	0,244	0,819	0,526	0,247	0,870	0,707	0,509	0,699	0,648	0,555	0,721	0,633	0,547	0,988	0,685	0,259	0,829	0,606	0,384
CstPW Scaled	PSNR	Avg	56,96	44,76	32,88	57,49	41,92	29,66	41,49	36,54	28,15	57,27	48,67	32,81	50,61	47,93	36,91	49,02	45,53	36,36	59,18	40,53	27,22	53,15	43,70	32,00
		Min	48,09	37,21	27,58	56,04	38,85	28,00	40,28	33,76	24,51	56,64	45,18	30,78	45,51	44,37	35,21	36,15	31,93	26,49	58,98	40,12	26,88	48,81	38,78	28,49
	Power	Avg	0,755	0,478	0,306	0,911	0,523	0,233	0,800	0,500	0,231	0,865	0,700	0,503	0,689	0,639	0,546	0,710	0,619	0,531	0,987	0,680	0,252	0,817	0,591	0,372
Short ModPW	PSNR	Avg	57,98	41,94	31,28	57,70	41,98	29,65	41,51	36,48	28,26	57,31	47,43	32,72	52,34	49,09	36,97	49,67	45,83	36,40	59,19	40,54	27,23	53,67	43,50	31,80
		Min	55,12	38,53	27,88	56,81	38,86	28,00	40,28	34,03	24,56	56,66	45,70	30,78	45,59	44,43	35,09	44,61	34,59	26,71	58,83	40,15	26,88	51,13	39,47	28,56
	Power	Avg	0,751	0,492	0,326	0,908	0,521	0,233	0,794	0,495	0,234	0,864	0,705	0,500	0,689	0,638	0,545	0,709	0,619	0,531	0,987	0,679	0,252	0,815	0,592	0,375
Long ModPW	PSNR	Avg	58,18	47,01	33,54	57,81	43,08	30,26	41,63	38,45	28,55	57,33	50,54	33,20	52,86	51,25	38,40	50,01	46,68	37,44	59,12	40,70	27,36	53,86	45,39	32,41
		Min	55,92	43,48	32,20	57,43	42,06	28,86	40,27	36,95	27,03	56,66	50,07	31,86	51,06	50,82	37,13	48,90	46,32	36,99	58,91	40,25	26,90	52,75	44,05	31,31
	Power	Avg	0,798	0,506	0,383	0,922	0,505	0,234	0,820	0,524	0,236	0,869	0,725	0,510	0,702	0,666	0,564	0,720	0,605	0,561	0,958	0,685	0,257	0,831	0,608	0,360

Fig. 6. Results in terms of PSNR and power of each video methods for each sequence (averaged over all the frames).

color, details, as well as temporal variation (motion and global luminance).[2]

The starting point for this experiment is the constant power-weight method (*CstPW*) described in Section V-A, which represents the best possible solution at any chosen trade-off between quality and power. Three power weights values were used for the *CstPW* method: $q \in \{1, 100, 400\}$. They correspond respectively to the second, fourth and last points on Fig. 2(a) and 2(b). The power consumptions obtained by those power weights in Section IV-C are 0.35, 0.55, and 0.7, with corresponding PSNR levels of 35 dB, 45 dB, and 56 dB. They represent three realistic settings for a TV set: high-power saving (a "green" setting), trade-off between power saving and quality and high quality. The PSNR (with 10-bit quantization of the LC values) as a function of power obtained with those power weights by the *CstPW* method are shown in Fig. 5. The results are averaged over all frames of each sequence and the standard deviation between frames is shown through error bars (vertical for PSNR and horizontal for power).

The methods compared in this experiment are:

- the constant quality/power trade-off mode (*CstPW*);
- the constant quality/power trade-off mode with additional scaling of the LED (*CstPW Scaled*) when the *MPC* is exceeded. The maximum power is here defined as the average power obtained with the *CstPW* method;
- the Short-term power constrained mode (*Short ModPW*) that starts from the *CstPW* mode, then scales LEDs and update the power weight when the *MPC* is exceeded (Section V-C). The *MPC* is here defined as the average power obtained with *CstPW* method;
- the Long-term power constrained mode (*Long ModPW*) from Section V-D with a *Threshold Buffer* $T_B = 0.05$. The chosen T_B is smaller than a realistic setting but fitted for an experiment on 100-frames-long sequences.

2) Results for Each Videos: Fig. 6 presents PSNR and power consumption for each of the tested methods and each sequence. The *CstPW* method has a highly varying power consumption but a fairly stable quality. Inversely, scaling directly the LED values (*CstPW Scaled* method) amounts to fully choosing power consumption over quality and can cause significant degradations, as shown by the decreases in the minimum obtained PSNR. Modifying the power weight (*Short ModPW*) allows to follow drastic power restrictions while maintaining the best possible quality. Globally, each method corresponds to a choice between variations in power or in quality. The more tightly the power consumption is constrained, the less it varies. It consequently implies that quality varies more and reaches significantly lower values.

Fig. 7 shows the evolution of PSNR and power over time for the presented methods for two selected examples. It illustrates how constraining power consumption influences the quality, depending on the chosen method.

[2] A frame extracted from each of them is available as supplemental material, VideosDatabase.png, online at http://ieeexplore.ieee.org

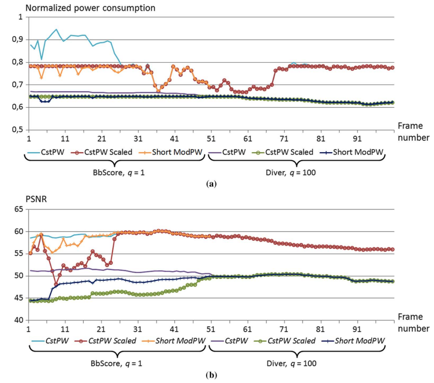


Fig. 7. Examples of the evolution of power (top) and PSNR (bottom) depending on the frame number for the BbScore and Fireworks sequences respectively at power weights of 1 and 100.

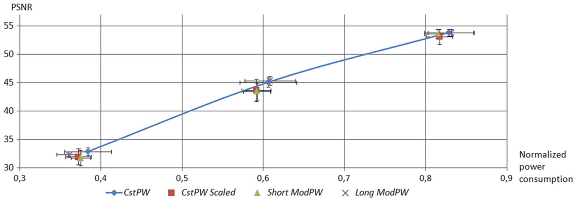


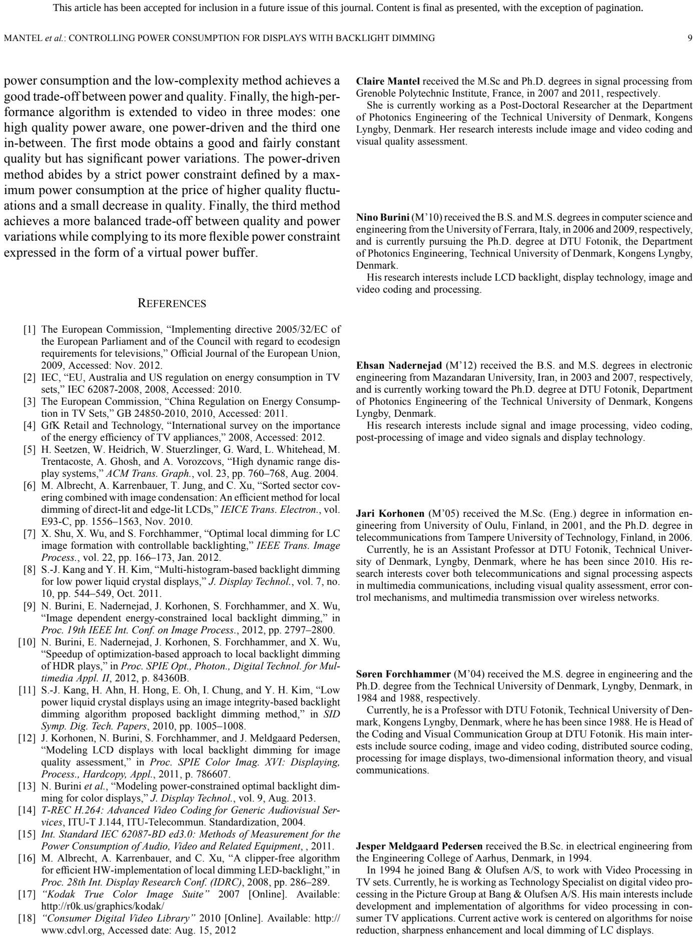
Fig. 8. PSNR as a function of power averaged over all sequences for the presented methods. The standard deviation of PSNR and power over the frames of a sequence are represented through the errors bars.

3) Global Results: Fig. 8 shows the PSNR(power) curves for each of the presented methods, averaged over all frames and all sequences. The error bars in this figure represent the variation of quality and power over time (i.e. between frames).

Those global results are in agreement with the results detailed for each sequence even though averaging over all frames has diminished differences. The characteristics of each method are especially highlighted through the error bars. The *CstPW* method that favors quality has the highest overall quality and the lowest quality standard deviation, but this method has also the largest standard deviation for power. The *CstPW Scaled* and *Short ModPW* methods obtain the smallest power variation but they conversely have the highest quality fluctuations. Finally the *Long ModPW* method has power and quality similar to *CstPW* but presents less differences between quality and power consumption variation.

VI. Conclusion

The increasing awareness of consumers to energy consumption in electronics and the evolution of power consumption regulations have made power consumption one of the major issues for TV sets. Backlight dimming is an important technique to achieve power savings in LCD sets. This paper presents a global framework for a backlight power management system. High-performance and low-complexity approaches are then introduced, both providing means to achieve any target power level. Compared to algorithms from the literature, the high-performance method achieves the best possible quality for a given



power consumption and the low-complexity method achieves a good trade-off between power and quality. Finally, the high-performance algorithm is extended to video in three modes: one high quality power aware, one power-driven and the third one in-between. The first mode obtains a good and fairly constant quality but has significant power variations. The power-driven method abides by a strict power constraint defined by a maximum power consumption at the price of higher quality fluctuations and a small decrease in quality. Finally, the third method achieves a more balanced trade-off between quality and power variations while complying to its more flexible power constraint expressed in the form of a virtual power buffer.

REFERENCES

[1] The European Commission, "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, 2009, Accessed: Nov. 2012.

[2] IEC, "EU, Australia and US regulation on energy consumption in TV sets," IEC 62087-2008, 2008, Accessed: 2010.

[3] The European Commission, "China Regulation on Energy Consumption in TV Sets," GB 24850-2010, 2010, Accessed: 2011.

[4] GfK Retail and Technology, "International survey on the importance of the energy efficiency of TV appliances," 2008, Accessed: 2012.

[5] H. Seetzen, W. Heidrich, W. Stuerzlinger, G. Ward, L. Whitehead, M. Trentacoste, A. Ghosh, and A. Vorozcovs, "High dynamic range display systems," *ACM Trans. Graph.*, vol. 23, pp. 760–768, Aug. 2004.

[6] 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 Trans. Electron.*, vol. E93-C, pp. 1556–1563, Nov. 2010.

[7] X. Shu, X. Wu, and S. Forchhammer, "Optimal local dimming for LC image formation with controllable backlighting," *IEEE Trans. Image Process.*, vol. 22, pp. 166–173, Jan. 2012.

[8] 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.

[9] N. Burini, E. Nadernejad, J. Korhonen, S. Forchhammer, and X. Wu, "Image dependent energy-constrained local backlight dimming," in *Proc. 19th IEEE Int. Conf. on Image Process.*, 2012, pp. 2797–2800.

[10] N. Burini, E. Nadernejad, J. Korhonen, S. Forchhammer, and X. Wu, "Speedup of optimization-based approach to local backlight dimming of HDR plays," in *Proc. SPIE Opt., Photon., Digital Technol. for Multimedia Appl. II*, 2012, p. 84360B.

[11] S.-J. Kang, H. Ahn, H. Hong, E. Oh, I. Chung, and Y. H. Kim, "Low power liquid crystal displays using an image integrity-based backlight dimming algorithm proposed backlight dimming method," in *SID Symp. Dig. Tech. Papers*, 2010, pp. 1005–1008.

[12] J. Korhonen, N. Burini, S. Forchhammer, and J. Meldgaard Pedersen, "Modeling LCD displays with local backlight dimming for image quality assessment," in *Proc. SPIE Color Imag. XVI: Displaying, Process., Hardcopy, Appl.*, 2011, p. 786607.

[13] N. Burini *et al.*, "Modeling power-constrained optimal backlight dimming for color displays," *J. Display Technol.*, vol. 9, Aug. 2013.

[14] *T-REC H.264: Advanced Video Coding for Generic Audiovisual Services*, ITU-T J.144, ITU-Telecommun. Standardization, 2004.

[15] *Int. Standard IEC 62087-BD ed3.0: Methods of Measurement for the Power Consumption of Audio, Video and Related Equipment*, , 2011.

[16] M. Albrecht, A. Karrenbauer, and C. Xu, "A clipper-free algorithm for efficient HW-implementation of local dimming LED-backlight," in *Proc. 28th Int. Display Research Conf. (IDRC)*, 2008, pp. 286–289.

[17] *"Kodak True Color Image Suite"* 2007 [Online]. Available: http://r0k.us/graphics/kodak/

[18] *"Consumer Digital Video Library"* 2010 [Online]. Available: http://www.cdvl.org, Accessed date: Aug. 15, 2012

Claire Mantel received the M.Sc and Ph.D. degrees in signal processing from Grenoble Polytechnic Institute, France, in 2007 and 2011, respectively.

She is currently working as a Post-Doctoral Researcher at the Department of Photonics Engineering of the Technical University of Denmark, Kongens Lyngby, Denmark. Her research interests include image and video coding and visual quality assessment.

Nino Burini (M'10) received the B.S. and M.S. degrees in computer science and engineering from the University of Ferrara, Italy, in 2006 and 2009, respectively, and is currently pursuing the Ph.D. degree at DTU Fotonik, the Department of Photonics Engineering, Technical University of Denmark, Kongens Lyngby, Denmark.

His research interests include LCD backlight, display technology, image and video coding and processing.

Ehsan Nadernejad (M'12) received the B.S. and M.S. degrees in electronic engineering from Mazandaran University, Iran, in 2003 and 2007, respectively, and is currently working toward the Ph.D. degree at DTU Fotonik, Department of Photonics Engineering of the Technical University of Denmark, Kongens Lyngby, Denmark.

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 the M.Sc. (Eng.) degree in information engineering from University of Oulu, Finland, in 2001, and the Ph.D. degree in telecommunications from Tampere University of Technology, Finland, in 2006.

Currently, he is an Assistant Professor at DTU Fotonik, Technical University of Denmark, Lyngby, Denmark, where he has been 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.

Søren Forchhammer (M'04) received the M.S. degree in engineering and the Ph.D. degree from the Technical University of Denmark, Lyngby, Denmark, in 1984 and 1988, respectively.

Currently, he is a Professor with DTU Fotonik, Technical University of Denmark, Kongens Lyngby, Denmark, where he has been since 1988. He is Head of the Coding and Visual Communication Group at DTU Fotonik. His main interests include source coding, image and video coding, distributed source coding, processing for image displays, two-dimensional information theory, and visual communications.

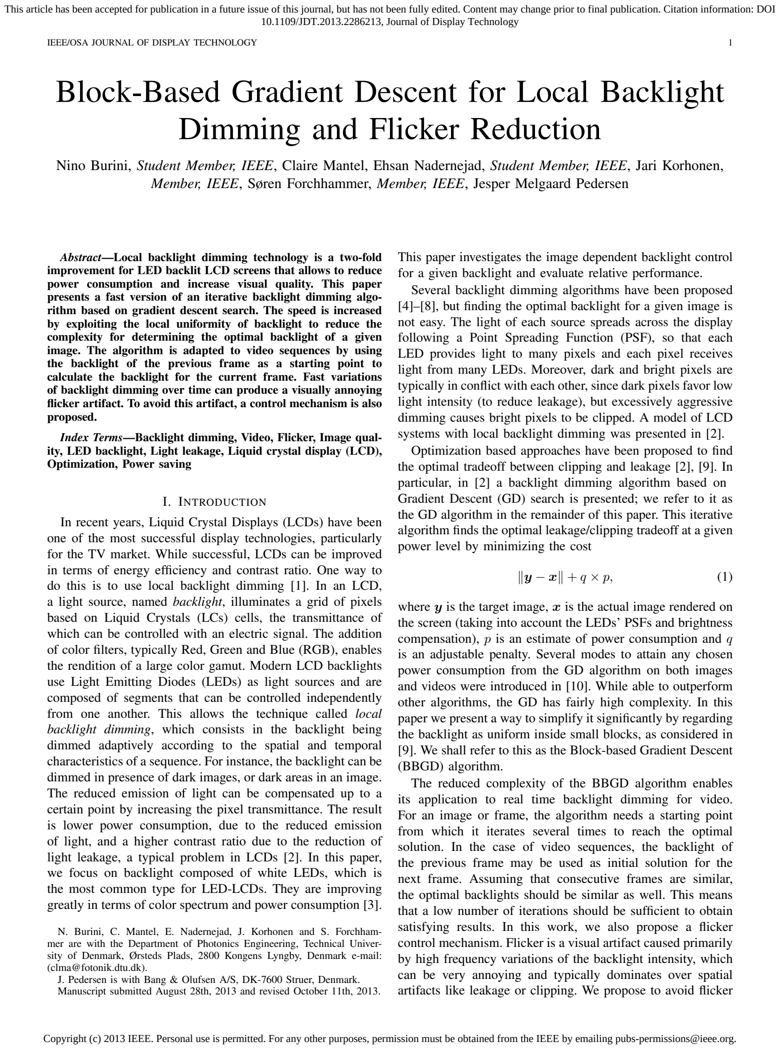
Jesper Meldgaard Pedersen received the B.Sc. in electrical engineering from the Engineering College of Aarhus, Denmark, in 1994.

In 1994 he joined Bang & Olufsen A/S, to work with Video Processing in TV sets. Currently, he is working as Technology Specialist on digital video processing in the Picture Group at Bang & Olufsen A/S. His main interests include development and implementation of algorithms for video processing in consumer TV applications. Current active work is centered on algorithms for noise reduction, sharpness enhancement and local dimming of LC displays.

A.12 Block-Based Gradient Descent for Local Backlight Dimming and Flicker Reduction

N. Burini, C. Mantel, **E. Nadernejad**, J. Korhonen, S. Forchhammer, and J. Pedersen, “Block-Based Gradient Descent for Local Backlight Dimming and Flicker Reduction”, *IEEE/OSA Journal of Display Technology*, vol. PP, no. 99, pp. 1–1, 2013. DOI: `10.1109/JDT.2013.2286213`

Reference: [12]



Block-Based Gradient Descent for Local Backlight Dimming and Flicker Reduction

Nino Burini, *Student Member, IEEE*, Claire Mantel, Ehsan Nadernejad, *Student Member, IEEE*, Jari Korhonen, *Member, IEEE*, Søren Forchhammer, *Member, IEEE*, Jesper Melgaard Pedersen

***Abstract*—Local backlight dimming technology is a two-fold improvement for LED backlit LCD screens that allows to reduce power consumption and increase visual quality. This paper presents a fast version of an iterative backlight dimming algorithm based on gradient descent search. The speed is increased by exploiting the local uniformity of backlight to reduce the complexity for determining the optimal backlight of a given image. The algorithm is adapted to video sequences by using the backlight of the previous frame as a starting point to calculate the backlight for the current frame. Fast variations of backlight dimming over time can produce a visually annoying flicker artifact. To avoid this artifact, a control mechanism is also proposed.**

***Index Terms*—Backlight dimming, Video, Flicker, Image quality, LED backlight, Light leakage, Liquid crystal display (LCD), Optimization, Power saving**

I. Introduction

In recent years, Liquid Crystal Displays (LCDs) have been one of the most successful display technologies, particularly for the TV market. While successful, LCDs can be improved in terms of energy efficiency and contrast ratio. One way to do this is to use local backlight dimming [1]. In an LCD, a light source, named *backlight*, illuminates a grid of pixels based on Liquid Crystals (LCs) cells, the transmittance of which can be controlled with an electric signal. The addition of color filters, typically Red, Green and Blue (RGB), enables the rendition of a large color gamut. Modern LCD backlights use Light Emitting Diodes (LEDs) as light sources and are composed of segments that can be controlled independently from one another. This allows the technique called *local backlight dimming*, which consists in the backlight being dimmed adaptively according to the spatial and temporal characteristics of a sequence. For instance, the backlight can be dimmed in presence of dark images, or dark areas in an image. The reduced emission of light can be compensated up to a certain point by increasing the pixel transmittance. The result is lower power consumption, due to the reduced emission of light, and a higher contrast ratio due to the reduction of light leakage, a typical problem in LCDs [2]. In this paper, we focus on backlight composed of white LEDs, which is the most common type for LED-LCDs. They are improving greatly in terms of color spectrum and power consumption [3]. This paper investigates the image dependent backlight control for a given backlight and evaluate relative performance.

Several backlight dimming algorithms have been proposed [4]–[8], but finding the optimal backlight for a given image is not easy. The light of each source spreads across the display following a Point Spreading Function (PSF), so that each LED provides light to many pixels and each pixel receives light from many LEDs. Moreover, dark and bright pixels are typically in conflict with each other, since dark pixels favor low light intensity (to reduce leakage), but excessively aggressive dimming causes bright pixels to be clipped. A model of LCD systems with local backlight dimming was presented in [2].

Optimization based approaches have been proposed to find the optimal tradeoff between clipping and leakage [2], [9]. In particular, in [2] a backlight dimming algorithm based on Gradient Descent (GD) search is presented; we refer to it as the GD algorithm in the remainder of this paper. This iterative algorithm finds the optimal leakage/clipping tradeoff at a given power level by minimizing the cost

$$\|\boldsymbol{y} - \boldsymbol{x}\| + q \times p, \tag{1}$$

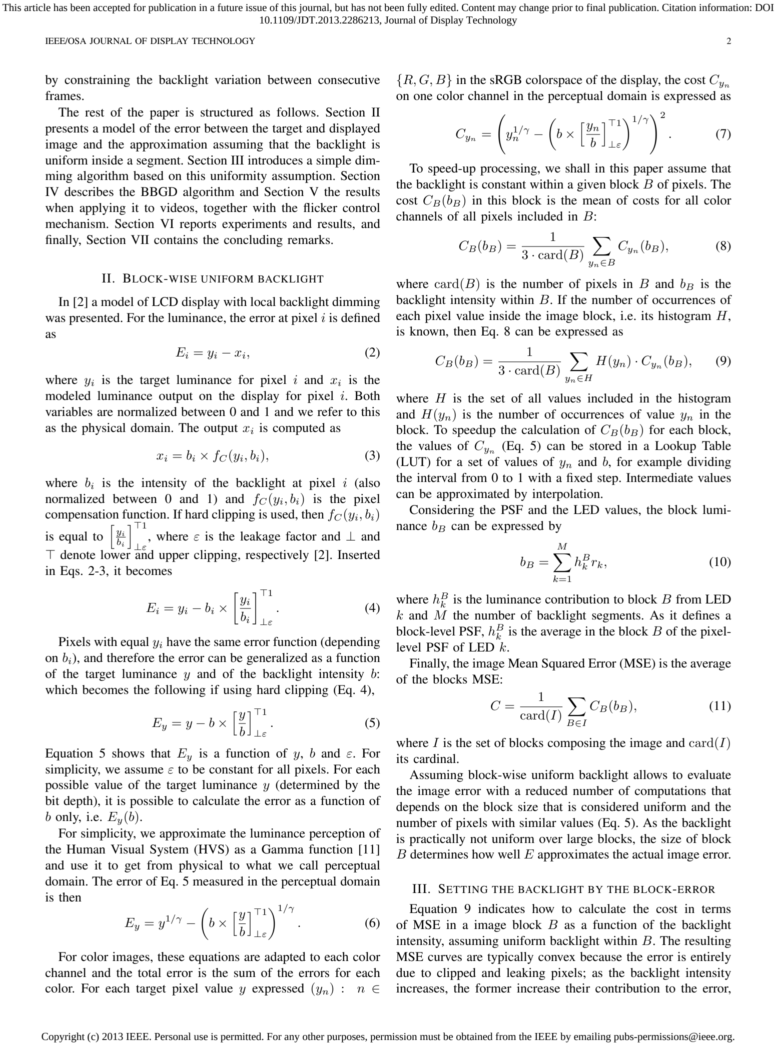
where $\boldsymbol{y}$ is the target image, $\boldsymbol{x}$ is the actual image rendered on the screen (taking into account the LEDs' PSFs and brightness compensation), p is an estimate of power consumption and q is an adjustable penalty. Several modes to attain any chosen power consumption from the GD algorithm on both images and videos were introduced in [10]. While able to outperform other algorithms, the GD has fairly high complexity. In this paper we present a way to simplify it significantly by regarding the backlight as uniform inside small blocks, as considered in [9]. We shall refer to this as the Block-based Gradient Descent (BBGD) algorithm.

The reduced complexity of the BBGD algorithm enables its application to real time backlight dimming for video. For an image or frame, the algorithm needs a starting point from which it iterates several times to reach the optimal solution. In the case of video sequences, the backlight of the previous frame may be used as initial solution for the next frame. Assuming that consecutive frames are similar, the optimal backlights should be similar as well. This means that a low number of iterations should be sufficient to obtain satisfying results. In this work, we also propose a flicker control mechanism. Flicker is a visual artifact caused primarily by high frequency variations of the backlight intensity, which can be very annoying and typically dominates over spatial artifacts like leakage or clipping. We propose to avoid flicker

N. Burini, C. Mantel, E. Nadernejad, J. Korhonen and S. Forchhammer are with the Department of Photonics Engineering, Technical University of Denmark, Ørsteds Plads, 2800 Kongens Lyngby, Denmark e-mail: (clma@fotonik.dtu.dk).

J. Pedersen is with Bang & Olufsen A/S, DK-7600 Struer, Denmark.

Manuscript submitted August 28th, 2013 and revised October 11th, 2013.



by constraining the backlight variation between consecutive frames.

The rest of the paper is structured as follows. Section II presents a model of the error between the target and displayed image and the approximation assuming that the backlight is uniform inside a segment. Section III introduces a simple dimming algorithm based on this uniformity assumption. Section IV describes the BBGD algorithm and Section V the results when applying it to videos, together with the flicker control mechanism. Section VI reports experiments and results, and finally, Section VII contains the concluding remarks.

II. Block-wise uniform backlight

In [2] a model of LCD display with local backlight dimming was presented. For the luminance, the error at pixel i is defined as

$$E_i = y_i - x_i, \tag{2}$$

where y_i is the target luminance for pixel i and x_i is the modeled luminance output on the display for pixel i. Both variables are normalized between 0 and 1 and we refer to this as the physical domain. The output x_i is computed as

$$x_i = b_i \times f_C(y_i, b_i), \tag{3}$$

where b_i is the intensity of the backlight at pixel i (also normalized between 0 and 1) and $f_C(y_i, b_i)$ is the pixel compensation function. If hard clipping is used, then $f_C(y_i, b_i)$ is equal to $\left[\frac{y_i}{b_i}\right]^{\top 1}_{\perp\varepsilon}$, where ε is the leakage factor and $\perp$ and $\top$ denote lower and upper clipping, respectively [2]. Inserted in Eqs. 2-3, it becomes

$$E_i = y_i - b_i \times \left[\frac{y_i}{b_i}\right]^{\top 1}_{\perp\varepsilon}. \tag{4}$$

Pixels with equal y_i have the same error function (depending on b_i), and therefore the error can be generalized as a function of the target luminance y and of the backlight intensity b: which becomes the following if using hard clipping (Eq. 4),

$$E_y = y - b \times \left[\frac{y}{b}\right]^{\top 1}_{\perp\varepsilon}. \tag{5}$$

Equation 5 shows that E_y is a function of y, b and ε. For simplicity, we assume ε to be constant for all pixels. For each possible value of the target luminance y (determined by the bit depth), it is possible to calculate the error as a function of b only, i.e. $E_y(b)$.

For simplicity, we approximate the luminance perception of the Human Visual System (HVS) as a Gamma function [11] and use it to get from physical to what we call perceptual domain. The error of Eq. 5 measured in the perceptual domain is then

$$E_y = y^{1/\gamma} - \left(b \times \left[\frac{y}{b}\right]^{\top 1}_{\perp\varepsilon}\right)^{1/\gamma}. \tag{6}$$

For color images, these equations are adapted to each color channel and the total error is the sum of the errors for each color. For each target pixel value y expressed $(y_n) : n \in \{R, G, B\}$ in the sRGB colorspace of the display, the cost C_{y_n} on one color channel in the perceptual domain is expressed as

$$C_{y_n} = \left(y_n^{1/\gamma} - \left(b \times \left[\frac{y_n}{b}\right]^{\top 1}_{\perp\varepsilon}\right)^{1/\gamma}\right)^2. \tag{7}$$

To speed-up processing, we shall in this paper assume that the backlight is constant within a given block B of pixels. The cost $C_B(b_B)$ in this block is the mean of costs for all color channels of all pixels included in B:

$$C_B(b_B) = \frac{1}{3 \cdot \mathrm{card}(B)} \sum_{y_n \in B} C_{y_n}(b_B), \tag{8}$$

where $\mathrm{card}(B)$ is the number of pixels in B and b_B is the backlight intensity within B. If the number of occurrences of each pixel value inside the image block, i.e. its histogram H, is known, then Eq. 8 can be expressed as

$$C_B(b_B) = \frac{1}{3 \cdot \mathrm{card}(B)} \sum_{y_n \in H} H(y_n) \cdot C_{y_n}(b_B), \tag{9}$$

where H is the set of all values included in the histogram and $H(y_n)$ is the number of occurrences of value y_n in the block. To speedup the calculation of $C_B(b_B)$ for each block, the values of C_{y_n} (Eq. 5) can be stored in a Lookup Table (LUT) for a set of values of y_n and b, for example dividing the interval from 0 to 1 with a fixed step. Intermediate values can be approximated by interpolation.

Considering the PSF and the LED values, the block luminance b_B can be expressed by

$$b_B = \sum_{k=1}^{M} h_k^B r_k, \tag{10}$$

where h_k^B is the luminance contribution to block B from LED k and M the number of backlight segments. As it defines a block-level PSF, h_k^B is the average in the block B of the pixel-level PSF of LED k.

Finally, the image Mean Squared Error (MSE) is the average of the blocks MSE:

$$C = \frac{1}{\mathrm{card}(I)} \sum_{B \in I} C_B(b_B), \tag{11}$$

where I is the set of blocks composing the image and $\mathrm{card}(I)$ its cardinal.

Assuming block-wise uniform backlight allows to evaluate the image error with a reduced number of computations that depends on the block size that is considered uniform and the number of pixels with similar values (Eq. 5). As the backlight is practically not uniform over large blocks, the size of block B determines how well E approximates the actual image error.

III. Setting the backlight by the block-error

Equation 9 indicates how to calculate the cost in terms of MSE in a image block B as a function of the backlight intensity, assuming uniform backlight within B. The resulting MSE curves are typically convex because the error is entirely due to clipped and leaking pixels; as the backlight intensity increases, the former increase their contribution to the error,

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JDT.2013.2286213, Journal of Display Technology

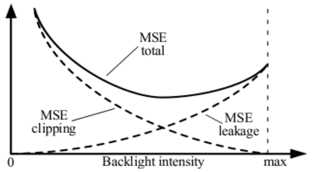


Fig. 1. Example of convex error curve caused by the contributions to the error by clipping and leakage.

while the latter decrease it. Summing these two contributions produces a convex curve with one minimum value, as shown in Fig. 1. It is therefore possible to determine what backlight intensity minimizes the error in each block.

Assuming uniform backlight blocks of similar size to that of the backlight segments, it is possible to design a dimming algorithm exploiting the block error (Eq. 9) by simply setting the LED intensity of each segment to the backlight intensity minimizing the MSE of the corresponding block. The effectiveness of this approach is reported in Sec. VI. Since the block size coincides with the segment size, we refer to this algorithm as the *segment-error* algorithm.

As in Eq. 1, we include an adjustable term in the segment error function, to penalize power consumption similarly to the GD algorithm [2], where the coefficient q determines the weight of the power penalty term. The relationship between LED intensity and energy consumption is linear [10] and the relative power consumption is $p = \sum_{k=1}^{M} r_k$.

This algorithm can be extended to block sizes smaller than the segment size. First, the backlight intensity minimizing the error in each block is calculated from the block error function. The LED intensity can then be set to the maximum of these values for the blocks contained within the segment.

IV. Block-based Gradient Descent

An existing backlight solution can be improved in terms of a given cost function f using GD search, as shown in [2]. The cost function depends on the error measured at pixel level, but we also included a term dependent on power consumption (Eq. 1). The assumption of block-wise uniform backlight can reduce the number of variables and speed up the algorithm. The cost function is then defined as

$$f = \sum_{B \in I} C_B(b_B) + q \times \left(\sum_{k=1}^{M} r_k \right), \qquad (12)$$

where I is the set of backlight blocks in the image and the cost in terms of MSE, C_B, is given in Eq. 9.

A. Backlight dimming by Gradient Descent

In [2], we presented a backlight dimming algorithm based on GD search [12]. Given a display with M local backlight segments, the set of LED intensities r_k (with $k = 1, \ldots, M$) can be considered as the coordinates of a specific backlight in the solution space. The solution space has M dimensions and the cost function f associates a cost to all possible sets of LED values. It is possible to calculate the gradient of the cost function at any point in the solution space. The gradient indicates the direction of highest increase of cost from the current solution, in terms of variation of each LED value r_k. Modifying the LEDs in the opposite direction instead produces 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 LED intensity vector is calculated by

$$\boldsymbol{r}_{j+1} = \boldsymbol{r}_j - s\nabla f(\boldsymbol{r}_j), \qquad (13)$$

where $\boldsymbol{r}_j$ is the solution for iteration j, ∇f is the gradient of the cost function and s the step size.

The size of s can be chosen adaptively and can be different for each iteration. Short steps guarantee a decrease in cost but require more iterations to reach the optimum, while large steps can speed up the process but risk to deviate from the optimum. The algorithms presented in this work use a step selection strategy based on golden section search [13]. This bisection method allows to iteratively find the minimum of a function inside a given interval with a given precision, ensuring fast convergence and limiting the number of function evaluations. The method uses a "probe" to evaluate the cost function inside the interval and to decide which sub-interval contains the minimum.

Given the gradient at the current solution and Eq. 13, the cost of the next solution $\boldsymbol{r}_{j+1}$ can be expressed as a function of s. The golden section search needs an interval in which to look for the best step. The left extreme of this interval is obviously $s = 0$ but choosing the right extreme is more complicated. All LED values are contained within the [0, 1] range and as the s value grows, the components of $\boldsymbol{r}_{j+1}$ will reach one of those extrema (0 or 1). A realistic approach is therefore to choose the right extreme as the s value when the first LED saturates:

$$\max(s)\ ,\ r_{j+1}^{k} \in [0{,}1]\ \forall k \in \{1, ..., M\} \qquad (14)$$

where r_{j+1}^{k} is the intensity of the kth LED at the iteration $j+1$. If $\boldsymbol{r}_j$ is close to the edges of the solution space then the search interval for s is relatively small and the cost reduction achievable with only one iteration is likely to be limited. On the other hand, using a bigger value of s would mean that some LEDs reach values below 0 or above 1. To keep the LEDs in the solution space the values of $\boldsymbol{r}_{j+1}$ are clipped in the interval 0 and 1, which amounts to settings the corresponding gradient components to 0. As the gradient changes, the shape of the cost function changes as well, and optimally a new gradient leading to a new iteration should be computed. However, to reduce the number of gradient calculations and still extend the search range for s, the right extreme value is chosen as s for which all LED saturate (reaching 0 if $\nabla f(\boldsymbol{r}_j^k) > 0$ or 1 otherwise). To make sure that the implied change in the cost function does not break the convexity constraint necessary for golden section search, we verify that the cost of the probe is lower

than that of the chosen right extreme before doing the actual golden section search. If it is not, then the probe becomes the right extreme, and a new probe is calculated (the same mechanism was implemented in [2]). The process is iterated until the minimum is reached with the chosen precision.

B. Exploiting block-wise uniformity

As mentioned in Sec. III, assuming the block-wise backlight uniformity reduces the number of variables. This is particularly interesting for an approach based on GD to find the optimal backlight, since it computes the image error many times to estimate the gradient and select the step size. The values of $E_B(b_B)$ for each block can be stored for quick access, accelerating the GD iterations. We shall refer to this version of the GD algorithm generalized to block-wise uniform backlight as the BBGD algorithm. Larger 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. A block size of 1 pixel yields the same result as the original GD algorithm. Section VI includes an experiment assessing the performance of the BBGD algorithm.

V. Block-based Gradient Descent for Video

The BBGD algorithm needs a starting point to iterate from and the closer this starting point is to the optimal solution, the faster it converges. This makes the BBGD an interesting option for video content. Indeed, excluding scene cuts, the consecutive frames of a video sequence tend to be quite similar and so should be the corresponding optimal backlights. Therefore, using the backlight values of the previous frame as starting point for BBGD is efficient as it is most likely close to the optimum for the current frame. We propose to extend the backlight dimming algorithm with a scene change detector, e.g. based on the multiple histograms of the current and the previous frames [14] and use a simpler algorithm to initialize the backlight when a new scene is detected.

Video standards have typical framerates of 25 Hz or 30 Hz, so real-time backlight dimming must compute at least 25 backlights per second. Therefore, the algorithm used to initialize a new sequence should be fast, as it computes the backlight from scratch. We propose to use the segment-error algorithm (Sec. III) to initialize the scene and the BBGD algorithm to provide an iterative solution for the following frames. For fast calculation, the number of iterations should also be limited.

Quick backlight variations might cause flicker artifact, which is visually very annoying [15], [16]. Therefore, we also propose a control mechanism to avoid flicker by constraining the step taken by the BBGD below an adaptive threshold.

A. Initialization

The BBGD applied to videos needs to be initialized for the first frame it is applied on. We have used the segment-error algorithm described in Sec. III, including the power penalty, to initialize the backlight of a new sequence without any additional computation.

B. Updating the backlight

Using the BBGD approach, after the first frame the backlight of the previous frame is used as starting backlight solution for the next frame $\boldsymbol{r}_0^{t+1}$

For each iteration, the backlight of frame $t+1$ is updated as follows:

$$\boldsymbol{r}_{k+1}^{t+1} = \boldsymbol{r}_k^{t+1} - s\nabla f(\boldsymbol{r}_k^{t+1}), \tag{15}$$

where k and $k+1$ refer to the iteration index and s is the step size, selected as explained in Sec. IV-A. After a certain number of iterations, the algorithm stops and sets the backlight for frame $t+1$. The final solution for $\boldsymbol{r}^{t+1}$ is used as the initial solution for $\boldsymbol{r}^{t+2}$ and the algorithm is applied again.

C. Flicker control

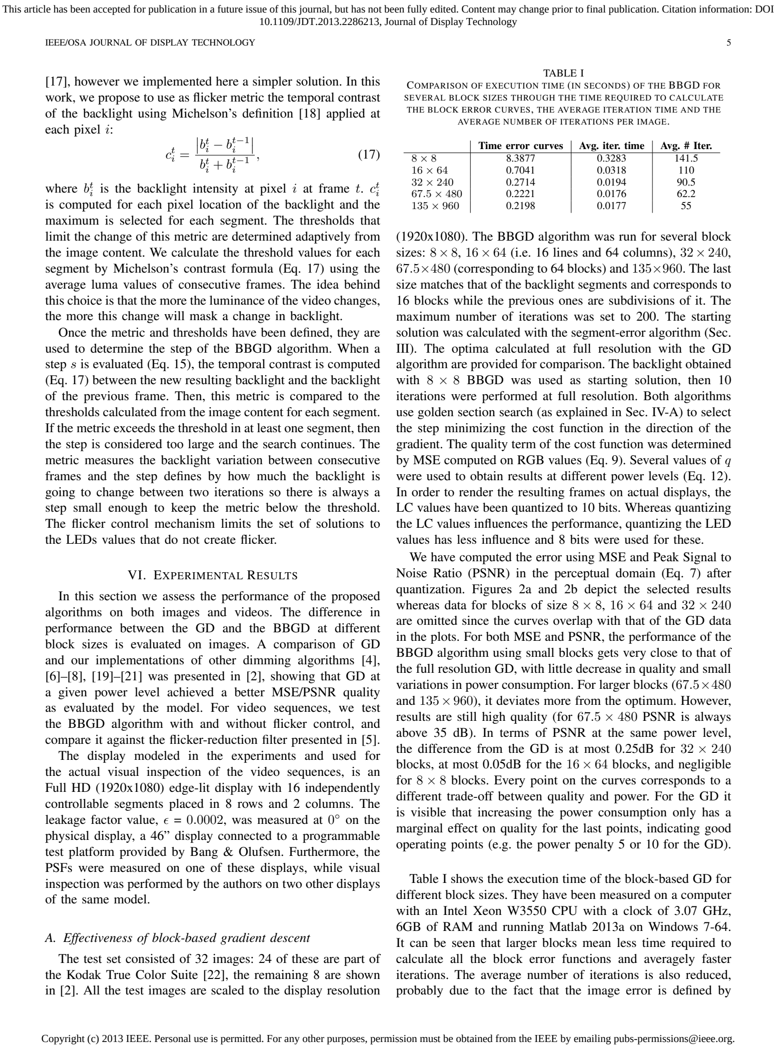
Backlight dimming of a video sequence vary over time accordingly to the video content. Variations of the backlight luminance above a certain frequency create visible flashes that are called flicker. This artifact is the main temporal defect of backlight dimming [15] and, due to its abruptness and short temporal duration. When present, it predominates over other backlight dimming defects. Several factors can cause flicker: abrupt changes in the backlight of consecutive frames (often occurring with algorithms that dim the backlight independently frame by frame and are sensitive to local changes), inaccuracies in the model (as inexact gamma or PSF) or in the synchronization of the LC cells and the LED backlight, or quantization of low LED values (which could mean a large relative backlight variation and cause a large change in the compensated transmittance).

While the other factors can be eliminated or reduced with accurate measurements and settings of the model, constraining the backlight variations needs flicker-aware algorithms. In [5], Chen et al. presented a backlight dimming algorithm featuring a temporal filter that smooths the temporal variation of the backlight and decreases flicker. On all LED values, they apply the adaptive first-order IIR filter defined by

$$\begin{aligned} \boldsymbol{r}_D^t &= R \cdot \boldsymbol{r}_D^{t-1} + (1-R) \cdot \boldsymbol{r}_C^t \\ R &= \min(1,\ 0.125 + \left|\text{mean}(p_i^t) - \text{mean}(p_i^{t-1})\right|, \end{aligned} \tag{16}$$

where $t+1$ is the current frame, p_i^t is the maximum value between the R, G and B channels at pixel i in frame t the r_C the LED values computed by the backlight dimming algorithm and r_D the filtered LED. While adaptive, this filter acts on the LED values, which does not guarantee complete flicker removal. Moreover, as Eq. 16 is applied as a causal post-processing, compared to a built-in approach it is prone to delayed reaction time. This can create significant clipping when an increase of LED intensity is reduced. As flicker reduction can introduce temporally varying clipping artifacts, clipping versus flicker tradeoffs should be optimized simultaneously.

We present a flicker reduction mechanism for the BBGD algorithm that limits inter-frame backlight variation by constraining the length of the step our algorithm can take. The constraint is defined as a threshold for a flicker metric specified for each backlight segment. The ideal flicker metric would be to use flicker perception thresholds and a flicker model [16],



[17], however we implemented here a simpler solution. In this work, we propose to use as flicker metric the temporal contrast of the backlight using Michelson's definition [18] applied at each pixel i:

$$c_i^t = \frac{\left|b_i^t - b_i^{t-1}\right|}{b_i^t + b_i^{t-1}}, \qquad (17)$$

where b_i^t is the backlight intensity at pixel i at frame t. c_i^t is computed for each pixel location of the backlight and the maximum is selected for each segment. The thresholds that limit the change of this metric are determined adaptively from the image content. We calculate the threshold values for each segment by Michelson's contrast formula (Eq. 17) using the average luma values of consecutive frames. The idea behind this choice is that the more the luminance of the video changes, the more this change will mask a change in backlight.

Once the metric and thresholds have been defined, they are used to determine the step of the BBGD algorithm. When a step s is evaluated (Eq. 15), the temporal contrast is computed (Eq. 17) between the new resulting backlight and the backlight of the previous frame. Then, this metric is compared to the thresholds calculated from the image content for each segment. If the metric exceeds the threshold in at least one segment, then the step is considered too large and the search continues. The metric measures the backlight variation between consecutive frames and the step defines by how much the backlight is going to change between two iterations so there is always a step small enough to keep the metric below the threshold. The flicker control mechanism limits the set of solutions to the LEDs values that do not create flicker.

VI. Experimental Results

In this section we assess the performance of the proposed algorithms on both images and videos. The difference in performance between the GD and the BBGD at different block sizes is evaluated on images. A comparison of GD and our implementations of other dimming algorithms [4], [6]–[8], [19]–[21] was presented in [2], showing that GD at a given power level achieved a better MSE/PSNR quality as evaluated by the model. For video sequences, we test the BBGD algorithm with and without flicker control, and compare it against the flicker-reduction filter presented in [5].

The display modeled in the experiments and used for the actual visual inspection of the video sequences, is an Full HD (1920x1080) edge-lit display with 16 independently controllable segments placed in 8 rows and 2 columns. The leakage factor value, ϵ = 0.0002, was measured at 0° on the physical display, a 46" display connected to a programmable test platform provided by Bang & Olufsen. Furthermore, the PSFs were measured on one of these displays, while visual inspection was performed by the authors on two other displays of the same model.

A. Effectiveness of block-based gradient descent

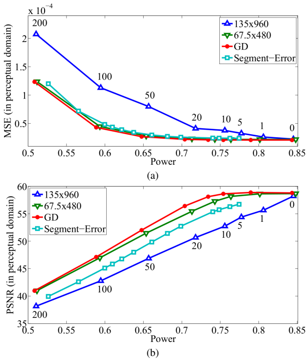
The test set consisted of 32 images: 24 of these are part of the Kodak True Color Suite [22], the remaining 8 are shown in [2]. All the test images are scaled to the display resolution (1920x1080). The BBGD algorithm was run for several block sizes: 8×8, 16×64 (i.e. 16 lines and 64 columns), 32×240, 67.5×480 (corresponding to 64 blocks) and 135×960. The last size matches that of the backlight segments and corresponds to 16 blocks while the previous ones are subdivisions of it. The maximum number of iterations was set to 200. The starting solution was calculated with the segment-error algorithm (Sec. III). The optima calculated at full resolution with the GD algorithm are provided for comparison. The backlight obtained with 8×8 BBGD was used as starting solution, then 10 iterations were performed at full resolution. Both algorithms use golden section search (as explained in Sec. IV-A) to select the step minimizing the cost function in the direction of the gradient. The quality term of the cost function was determined by MSE computed on RGB values (Eq. 9). Several values of q were used to obtain results at different power levels (Eq. 12). In order to render the resulting frames on actual displays, the LC values have been quantized to 10 bits. Whereas quantizing the LC values influences the performance, quantizing the LED values has less influence and 8 bits were used for these.

TABLE I
Comparison of execution time (in seconds) of the BBGD for several block sizes through the time required to calculate the block error curves, the average iteration time and the average number of iterations per image.

	Time error curves	Avg. iter. time	Avg. # Iter.
8×8	8.3877	0.3283	141.5
16×64	0.7041	0.0318	110
32×240	0.2714	0.0194	90.5
67.5×480	0.2221	0.0176	62.2
135×960	0.2198	0.0177	55

We have computed the error using MSE and Peak Signal to Noise Ratio (PSNR) in the perceptual domain (Eq. 7) after quantization. Figures 2a and 2b depict the selected results whereas data for blocks of size 8×8, 16×64 and 32×240 are omitted since the curves overlap with that of the GD data in the plots. For both MSE and PSNR, the performance of the BBGD algorithm using small blocks gets very close to that of the full resolution GD, with little decrease in quality and small variations in power consumption. For larger blocks (67.5×480 and 135×960), it deviates more from the optimum. However, results are still high quality (for 67.5×480 PSNR is always above 35 dB). In terms of PSNR at the same power level, the difference from the GD is at most 0.25dB for 32×240 blocks, at most 0.05dB for the 16×64 blocks, and negligible for 8×8 blocks. Every point on the curves corresponds to a different trade-off between quality and power. For the GD it is visible that increasing the power consumption only has a marginal effect on quality for the last points, indicating good operating points (e.g. the power penalty 5 or 10 for the GD).

Table I shows the execution time of the block-based GD for different block sizes. They have been measured on a computer with an Intel Xeon W3550 CPU with a clock of 3.07 GHz, 6GB of RAM and running Matlab 2013a on Windows 7-64. 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



(a)

(b)

Fig. 2. Average performance in terms of MSE (a) and PSNR (b), over the 32 test images, of the BBGD with several block sizes, the full resolution GD and the Segment-error algorithm. Curves for block sizes 8×8, 16×64 and 32×240 are not drawn as they overlap with the one for GD. The LC values are quantized to 10 bits. Labels indicate power penalty.

fewer elements (the blocks of uniform backlight). We wish to highlight that these results are dependent on software implementation and hardware, and that the code has not been optimized for high performance. However, they give a clear indication of how the block size affects execution time.

Overall, these preliminary results show that the BBGD 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.

B. Performance of the segment-error algorithm

The segment-error algorithm (Sec. III) was applied to the test images. This algorithm sets the LED intensities to the backlight intensity value that minimizes the block error defined by Eq. 9. Unless otherwise stated, the settings for this experiment are the same as in the previous one. The block size was set to 135×960 pixels, i.e. 16 blocks in all. The results are included in Figs. 2a and 2b (marked by □). This algorithm can achieve good quality results, even though the full resolution GD can achieve 2 to 3.5 dB more at the same power level. It is interesting to see that the Segment-error algorithm can obtain lower MSE (higher PSNR) than BBGD with 16 backlight blocks. This is contrary to our expectations, since BBGD iterations should improve the initial result provided by the Segment-error algorithm. Closer inspection has showed that BBGD with 16 blocks indeed obtains smaller error when using Eq. 11 to evaluate it. However, this computation assumes uniform backlight within each block and since this assumption is not accurate for such large blocks, the algorithm iterates using approximated results which explains this outcome.

Fig. 3. Sample frames of the sequences used for the experiments: (left to right, top to bottom) *BBBunny*, *Diver*, *Pedestrian*, *Stars* and *Volcano*.

C. Gradient descent for video

The main objective experiment was performed to study BBGD applied on video, as presented in Sec. V. The simulated display is the same used in the previous experiments. The leakage factor ε was set to 0.0002. The quality term of the cost function was given by MSE and the power penalty q was set to 1 (a relatively small value). The block size was set to 67.5×480 and 135×960 pixels, respectively. Five sequences of Full HD resolution (1920x1080p) were tested: *Volcano*, *Diver* and *BBBunny* (resp. 300, 300 and 250 frames, from [23]), *Pedestrian* (200 frames, from [24]) and *Stars* (157 frames, from [25]), a sample of each of them is shown in Fig. 3. The sequences were chosen to display various spatial characteristics and different motion: low (*Volcano*), average (*Diver*, *Pedestrian*) and high (*Stars*, *BBBunny*). The backlight of the first frame was calculated using the segment-error algorithm (see Sec. III), while the other frames were calculated with the BBGD algorithm with 64 blocks taking the backlight of the previous frame as starting solution. For each frame, the algorithm performed a fixed number of iterations, ranging from 1 to 3. For comparison, the optimum MSE with $q = 1$ was calculated for each frame using the BBGD algorithm with block size 8×8 (which showed to provide nearly optimal results), with the same conditions of the experiment described in Sec. VI-A.

Figure 4 shows the variation of PSNR and power consumption over time for the *Volcano* and *BBBunny* sequences. The BBGD algorithm with blocks of size 67.5×480, that is 64 blocks of uniform backlight, manages to follow the optimum with a slight PSNR decrease. In at least two occasions (at the beginning and at the end of the sequence), using more iterations slightly improves image quality. The results are analogous in the case of block of size 135×960 (16 blocks of uniform backlight), but the difference in PSNR is increased. The two sequences *Volcano* and *BBBunny* display respectively low and high motion, which only produces punctual differences on power and on quality.

Using two or three iterations gives small benefits also

TABLE II
AVERAGE PSNR, MSE AND POWER CONSUMPTION $q = 1$ OVER ALL FRAMES OF THE 5 TEST SEQUENCES FOR THE BBGD ALGORITHM WITH VARIOUS BLOCK SIZES AND DIFFERENT NUMBER OF ITERATIONS.

	PSNR	MSE	Power
8 × 8 (200 it.)	50.36	5.020 E-5	0.6486
135 × 960 (1 it.)	48.83	5.402 E-5	0.6798
135 × 960 (2 it.)	48.70	5.394 E-5	0.6759
135 × 960 (3 it.)	48.70	5.408 E-5	0.6755
67.5 × 480 (1 it.)	50.17	5.137 E-5	0.6699
67.5 × 480 (2 it.)	50.15	5.117 E-5	0.6649
67.5 × 480 (3 it.)	50.09	5.113 E-5	0.6632

for the *Diver*, *BBBunny* and *Stars* sequences, but not for *Pedestrian*, where the opposite is observed. Table II reports the PSNR, MSE and power consumption averaged over all frames of all sequences. Because of the *Pedestrian* sequence, MSE increases with 3 iterations on 16 blocks. Smaller blocks provide a consistently higher quality, confirming the results of Sec. VI-A. Using 64 blocks seems a good trade-off between speed and performance. More iterations often give a better result, although this is not always the case. The main reason (already explained in Sec. IV-B) is that the BBGD algorithm iterates assuming a uniform backlight, which is not always correct for large blocks sizes, causing iterations to non-optimal results. Another cause is that, when considering backlight of video sequences, the optimum changes gradually from frame to frame, like a *'moving target'*. Using only one iteration might not get as close to the current optimum as more iterations, but it may provide a better starting solution for the following frame.

Two of the authors also performed a visual inspection of the processed videos on the 46" edge-lit display. The LC and LED values were calculated off-line, but displayed real-time on the edge-lit screen using a special playback mode implemented for the test. All sequences presented flicker for every number of iterations. No conclusion could be drawn on the relation between the number of iterations and the appearance of flicker from this visual inspection. In some cases, flicker seems to be lower with one iteration and increase with two or three, but in others the opposite was observed. Sometimes the flicker appeared less intense, but more widely distributed spatially. Also, using 64 blocks instead of 16 did not seem to provide any clear benefit in terms of flicker. However, the general impression was that the optimal backlight (calculated with 8×8 blocks) appeared to have more flicker than the 135×960 or 67.5×480 blocks. To conclude, the benefits of using more than one iteration appear limited and not consistent. Moreover, the number of iterations does not affect significantly the general level of flicker.

D. Effectiveness of flicker control

As noted in the previous section, all the generated videos presented flicker. Therefore the flicker control mechanism described in Sec. V-C was applied and tested on the same dataset with a block size of 67.5×480 and 1 iteration. For comparison, the flicker-reduction filter from Chen et al. [5] was applied on the BBGD results without flicker control.

Figures 5 and 6 show some results for the *BBBunny* sequence. Performance results averaged over all sequences are shown in Table III. Enabling flicker control can cause PSNR to decrease punctually, but on average the power and quality vary only marginally. It can be seen that the maximum temporal contrast and the maximum change in backlight luminance are smaller. This is expected since the metric is limited by the flicker control. Both LED values and the resulting backlight change more smoothly. Visual inspection showed that the flicker, clearly present when flicker control is disabled, is now removed on all sequences when the mechanism is enabled. With Chen's filter the flicker was significantly reduced but still present on the sequences *Pedestrian* and *BBBunny*. One advantage of the built-in mechanism compared to a post-processing filter can be seen in Fig. 5 around frame number 50: the flicker control mechanism detects that there is no risk of flicker so the LEDs follow that of the version without flicker-control, whereas the Chen filter is active anyway, resulting in a local temporary increase in error.

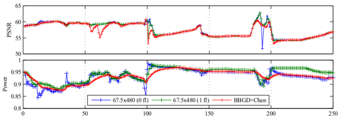


Fig. 5. PSNR and power consumption achieved by the BBGD algorithm on the *BBBunny* sequence with 64 blocks of uniform backlight with 1 iteration per frame with flicker control enabled/disabled compared against the Chen filter [5] applied on the BBGD with no flicker control.

TABLE III
PSNR, MSE, POWER, MAXIMUM CHANGE IN BACKLIGHT LUMINANCE (*Max BL Change*) AND MAXIMUM TEMPORAL CONTRAST AVERAGED OVER ALL FRAMES OF THE 5 TEST SEQUENCES. THE BBGD PARAMETERS ARE $q = 1$, A BLOCK SIZE OF 67.5×480 AND 1 ITERATION.

	BBGD	BBGD FC	BBGD + Chen
PSNR	50.88	50.54	50.81
MSE	5.038 E-5	6.18 E-5	5.2 E-5
Power consumption	0.638	0.648	0.638
Max BL Change	0.038	0.008	0.014
Max Tmp Contrast	0.049	0.014	0.021

VII. CONCLUSION

In this paper, we have presented two extensions of our backlight dimming algorithm based on gradient descent to achieve a fast flicker-free solution. Speed-up is achieved by approximating the backlight luminance as uniform within a block, enabling fast GD calculations. This provides a considerable reduction of the execution time without significant decrease in performance in terms of quality or power consumption. Approximating the backlight with 64 uniform blocks provides a good trade-off of speed and accuracy. Experiments on a test set of 5 videos showed significantly faster processings for an average PSNR degradation of less than 1dB. Applying this extension to video shows that a single iteration on each new frame is sufficient, as the biggest improvement happens with the first iteration. This block-based algorithm (BBGD) with a single iteration enables real-time implementation of our algorithm to video.

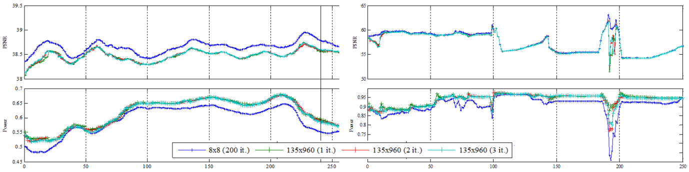


Fig. 4. PSNR and power consumption achieved by the BBGD algorithm on low and high motion sequences: *Volcano* (left) and *BBBunny* (right) sequence with 64 blocks of uniform backlight with 1, 2 or 3 iterations per frame, compared against the optimal result calculated on blocks with a size of 8 × 8 pixels. The horizontal axes is the frame number.

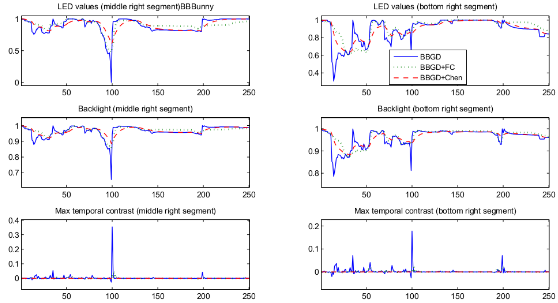


Fig. 6. LED values, average backlight intensity and maximum temporal contrast of the two bottom segments for the *BBBunny* sequence for the BBGD algorithm with 64 blocks with and without flicker control compared to the temporal filter by Chen et al. [5].

The second extension is the integration of a control process to avoid changes of the backlight luminance over time that are too large and thereby create flicker artifacts. It is based on limiting the temporal contrast of the LEDs backlight with respect to a threshold chosen adaptively based on the video content. Objective measures show that this flicker control decreases the maximum backlight change over time, while maintaining the average quality and power performance. Visual inspection on a test display verified that it avoided flicker. However, the flicker metric would benefit from being investigated more thoroughly.

REFERENCES

[1] M. Anandan, "Progress of LED backlights for LCDs," *J. Soc. Inf. Display*, vol. 16, no. 2, pp. 287–310, Feb 2008.

[2] 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*, vol. 9, no. 8, pp. 656–665, Aug 2013.

[3] M. Crawford, "Leds for solid-state lighting: Performance challenges and recent advances (invited paper)," *IEEE Journal of Selected Topics in Quantum Electronics*, vol. 15, no. 4, pp. 1028–1040, 2009.

[4] 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, Aug 2004.

[5] H. Chen, J. Sung, T. Ha, and Y. Park, "Locally pixel-compensated backlight dimming on LED-backlit LCD TV," *J. Soc. Inf. Display*, vol. 15, no. 12, p. 981, Dec 2007.

[6] 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, May 2009.

[7] 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. E93C, no. 11, pp. 1556–1563, Nov 2010.

[8] 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.

[9] X. Shu, X. Wu, and S. Forchhammer, "Optimal Local Dimming for LC Image Formation with Controllable Backlighting," *IEEE Trans. Image*

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JDT.2013.2286213, Journal of Display Technology

Process., vol. 22, no. 1, pp. 166–173, Jan 2013.

[10] C. Mantel, N. Burini, E. Nadernejad, J. Korhonen, S. Forchhammer, and J. M. Pedersen, "Controlling Power Consumption for Displays With Backlight Dimming," *IEEE/OSA Journal of Display Technology*, May 2013.

[11] C. Poynton, *A Technical Introduction to Digital Video*. John Wiley & Sons, 1996, ch. 6, p. 95.

[12] A.-L. Cauchy, "Méthode générale pour la résolution des systèmes d'équations simultanées," *Comp. Rend. Sci. Paris*, no. 25, pp. 536–538, Oct 1847.

[13] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C*, 2nd ed. Cambridge University Press, 1992.

[14] S.-J. Kang, S. I. Cho, S. Yoo, and Y.-H. Kim, "Scene change detection using multiple histograms for motion-compensated frame rate up-conversion," *IEEE/OSA Journal of Display Technology*, vol. 8, no. 3, pp. 121–126, March 2012.

[15] H. Chen, T. H. Ha, J. Sung, H. R. Kim, and B. H. Han, "Evaluation of LCD local-dimming-backlight system," *J. Soc. Inf. Display*, vol. 18, no. 1, p. 57, Jan 2010.

[16] J. D. Bullough, K. S. Hickcox, T. R. Klein, and N. Narendran, "Effects of flicker characteristics from solid-state lighting on detection, acceptability and comfort," *Lighting Research and Techhnology*, vol. 43, no. 3, pp. 337–348, 2011.

[17] A. B. Watson, *Handbook of Perception and Human Performance*. New York: Wiley, 1986, ch. Temporal Sensitivity.

[18] A. Michelson, "Studies in optics," Ph.D. dissertation, University of Chicago, 1927.

[19] 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, June 2008.

[20] 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, Dec 2009.

[21] S.-J. Kang and Y. H. Kim, "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.

[22] "Kodak True Color Image Suite," http://r0k.us/graphics/kodak/.

[23] *Consumer Digital Video Library*, $http://www.cdvl.org$, 2010.

[24] "Technische Universitaet Muenchen test sequences," ftp://ftp.ldv.e-technik.tu-muenchen.de/pub/test_sequences/.

[25] "Sita Sings the Blues," http://archive.org/details/Sita_Sings_the_Blues.

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.

Claire Mantel received the M.S. and Ph.D. degrees in signal processing from Grenoble Polytechnic Institute, France, in 2007 and 2011, respectively. She is currently working as a Post-Doctoral Researcher at the Department of Photonics Engineering of the Technical University of Denmark, Kongens Lyngby, Denmark. Her research interests include image and video coding and visual quality assessment.

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.

Søren 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.

Jesper Meldgaard Pedersen received the B.Sc. in electrical engineering from the Engineering College of Aarhus, Denmark, in 1994. In 1994 he joined Bang & Olufsen A/S, to work with Video Processing in TV sets. Currently, he is working as Technology Specialist on digital video processing in the Picture Group at Bang & Olufsen A/S. His main interests include development and implementation of algorithms for video processing in consumer TV applications. Current active work is centered on algorithms for noise reduction, sharpness enhancement and local dimming of LC displays.

A.13 How to Evaluate Objective Video Quality Metrics Reliably

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

Reference: [13]

HOW TO EVALUATE OBJECTIVE VIDEO QUALITY METRICS RELIABLY

Jari Korhonen[1], *Nino Burini*[1], *Junyong You*[2], and *Ehsan Nadernejad*[1]

[1]Dept. of Photonics Engineering
Technical University of Denmark (DTU)
Kgs. Lyngby, Denmark

[2]Dept. of Electronics and Telecommunications
Norwegian Univ. of Science and Tech. (NTNU)
Trondheim, Norway

ABSTRACT

The typical procedure for evaluating the performance of different objective quality metrics and indices involves comparisons between subjective quality ratings and the quality indices obtained using the objective metrics in question on the known video sequences. Several correlation indicators can be employed to assess how well the subjective ratings can be predicted from the objective values. In this paper, we give an overview of the potential sources for uncertainties and inaccuracies in such studies, related both to the method of comparison, possible inaccuracies in the subjective data, as well as processing of subjective data. We also suggest some general guidelines for researchers to make comparison studies of objective video quality metrics more reliable and useful for the practitioners in the field.

Index Terms— Video quality assessment, Objective metrics, Reliability

1. INTRODUCTION

Quality assessment is an essential part of evaluating different algorithms and methods for compression and transmission of digital video. However, quality is by definition a highly subjective feature influenced not only by the intrinsic characteristics of the signal, but also many psychological and environmental factors, which makes objective (i.e. algorithm based) measurement of quality a very challenging task. An ideal quality metric would predict the average opinion of the quality accurately, but both subjective quality assessment and mapping between objective data and subjective opinions have shown to be challenging [1-3].

Figure 1 outlines the typical procedure of evaluating the performance of an objective quality metric [4-5]. Test video sequences from an annotated video quality database with known subjective quality scores, usually referred to as mean opinion score (MOS), are used as an input for the evaluated objective quality assessment algorithm. Then, regression analysis is used to find a function that maps the obtained quality indices into subjective quality scores. This phase is necessary, since the range of the objective quality index and the subjective scores is usually not the same. Different prediction functions can be used for mapping; Video Quality Experts Group (VQEG) used a four parameter logistics function in their full reference objective metric validation study [6], but also linear, polynomial and exponential functions have been used for this purpose in the literature.

In the final phase, correlation analysis is performed to estimate how accurately the subjective scores are predicted from the objective quality indices. Root mean squared error (RMSE), Pearson correlation coefficient (PCC), Spearman rank order correlation coefficient (SROCC) and outlier ratio are the most commonly used indicators to evaluate the prediction accuracy, monotonicity, and consistency of a quality metric against the subjective quality data [4-5].

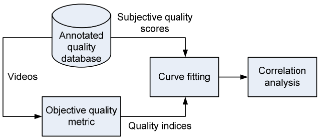


Figure 1. Performance evaluation of an objective metric.

There are numerous different factors impacting the reliability of performance analysis such as described above. The sources of inaccuracy may be related to the reliability of the subjective reference data, ie. the "ground truth", selection of video content, mapping between the subjective and objective quality values, and even the use and interpretation of the correlation indicators. In this paper, we explore these factors, with some practical examples.

The rest of the paper is organized as follows. In Section 2, we discuss the reliability of the subjective data. In Section 3, we explore the essentials for selecting the relevant data. In Section 4, we study the impact of different mapping functions in curve fitting. In Section 5, we discuss the methods for correlation analysis. Finally, the co-impact of

978-1-4673-0726-0/12/$31.00 © 2012 IEEE

different factors is discussed in Section 6, and the concluding remarks are given in Section 7.

2. RELIABILITY OF SUBJECTIVE DATA

We cannot expect reliable results for the assessment of objective metrics, if the subjective reference data itself is not reliable. The potential origins of uncertainty may be related to the methodology, environment and human psychology. It can be assumed that the influence of these factors can be minimized by following carefully the standardized recommendations for preparing the test environment, selecting the test methodology, and processing the data appropriately to compensate for personal bias and remove data from unreliable test subjects. However, when an external annotated database is used, it may be difficult to evaluate the reliability of the data. This is why we will summarize some typical sources of unreliability and suggest some approaches to verify the usability of an annotated quality database provided by a third party.

2.1 Sources of Unreliability

There is a multitude of different subjective quality assessment methods proposed in the literature, and many of them have even been standardized [7,8]. The methods can be classified in 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 MOS, that is the average of the scores by individual test subjects. MOS may be defined either using a continuous or discrete scale or a comparative distortion scale (*differential MOS*, DMOS), where different labels are assigned to disparate categories. Unfortunately, different test subjects may interpret vocabulary and intervals of the rating scale differently, depending on the language, instructions and individual psychological characteristics. Ratings may also be influenced by other factors, for example how interesting the content is [1]. This is why computation of MOS out of individual scores typically requires some post-processing, such as removal of outliers and systematic personal bias based on different interpretations of the rating scale [4, 9].

Pairwise comparisons and rank ordering are considered cognitively easier tasks than scoring. However, these methods carry some other problems. Most crucially, pairwise comparisons may require a large set of test cases to obtain a reliable result on a large number of sequences. On the other hand, the attention of the test subject may be distracted by several simultaneous stimuli, when rank ordering is performed. If the stimuli are not shown simultaneously but sequentially, it may be difficult for the test subjects to find subtle differences, in particular for sequences located temporally far from each other.

There are also potential causes of bias in the results that are relevant with several methodologies. For example, it has been shown that subjective scores are prone to a memory effect: distortions appearing just before grading have larger impact on the subjective opinion than those appearing earlier in the sequence [3]. Also the order of showing the sequences may impact the results [5]. This is why test sequences should ideally contain uniformly distributed distortions, and the sequences should be shown in a different pseudorandom order for each test subject. In this way, the potential bias based on the order or position of simultaneous stimuli can be compensated in the averaged results.

2.2 Evaluating the Reliability of Annotated Databases

In order to evaluate the reliability of an external quality database, the description of the test methodology should be studied carefully to find out how well the experiment takes the abovementioned potential sources of unreliability into account. However, even the best test methodologies cannot fully compensate for the uncertainty related to human factors, such as test subjects and consistency of instructions. In an ideal case, the subjective experiments would be repeated by different independent institutions. In this way, systematic errors could be compensated.

Unfortunately, it is expensive to arrange comprehensive large scale subjective quality assessment studies, and this is why multiple laboratories are most typically involved only in case of large test campaigns organized by established standardization bodies or experts groups. In VQEG subjective tests repeated by several laboratories [10], Pearson correlations between results by different laboratories range from 0.924 to 0.986, with mean of 0.97. EPFL-PoliMi database [9] has been created in collaboration with two institutes. The results from EPFL and PoliMi show a small systematic difference in quality scores, but in general, the results are reasonably well in line with each other, with correlations for different contents ranging from 0.9670 to 0.9945 (SROCC) and 0.9785 to 0.9938 (PCC) [9]. These results suggest that slightly less consistent MOS scores are obtained with databases containing several different types of distortions (VQEG) than more specific artifacts (EPFL-PoliMi). The correlations between subjective experiments on the same data by different laboratories could be considered as an upper limit for the correlations between subjective and objective data, that could be reached in case the objective metric is optimal.

If the resources do not allow repeating the original experiment, the reliability of a quality database can also be roughly evaluated from the standard deviation of the subjective evaluations, typically included in the results. It is also possible to arrange lightweight subjective validation tests, such as using pairwise comparisons or rank ordering for a small subset of the test data. If the comparison results are in line with the relative quality scores, it can be concluded that the database is reliable.

As an example, we have performed a small scale pairwise comparison study with sequences from the LIVE database [4,11]. We had 25 people of age 20 to 35 as test subjects, students or department staff with no experience in video quality assessment. The sequences have been displayed side by side on a 55-inch liquid crystal display, at a distance of about 2 times the screen height from the subjects. In order to focus in the potentially problematic cases, we have chosen pairs so that the mean subjective score shows different preference than some or all of the objective results from our previous study [5].

In the LIVE database, the subjective scores are expressed in terms of differential MOS (DMOS), where the scale is from 0 (best) to 100 (worst). In most of our test cases, DMOS values were close to each other, and the preferences between sequences are also relatively evenly distributed, suggesting that the visible differences in those sequences are very subtle indeed, just as DMOS indicates, and the preference was chosen on a random basis.

However, a couple of more interesting test cases were identified. Sequences sf4 and sf7 have relatively large difference in terms of DMOS (56.86 vs. 44.81), but in terms of preferences there is essentially a draw (13 vs. 12), indicating very similar perceived quality. Another case of interest is sf8 against sf13; DMOS favors sf8 (39.11 vs. 40.99), but pairwise comparisons favor sf13 with 22 against 3. For the record, the sequence sf ("Sunflower") shows a bee moving on a sunflower; sf4 is affected by wireless distortion, sf7 and sf8 by IP distortions and sf13 by MPEG-2 distortion.

The small number of tested sequence pairs do not allow us to make definitive conclusions about the reliability of the subjective data for the LIVE database. However, it reveals the fundamental difficulty of comparing sequences with different types of distortions: the pairs with unexpected comparison results contain fundamentally different artifacts. Also the content matters, since the visibility of different artifacts is greatly influenced by the level of details, area of interest, motion etc. Apparently, "Sunflower" represents especially challenging content from the perspective of subjective quality assessment.

3. SELECTION OF TEST DATA

It is essential that the used quality database contains video sequences with similar distortions for which the tested objective indices are designed. For example, peak signal-to-noise ratio (PSNR) is very sensitive to certain types of artifacts, such as spatial shift or contrast changes, but it has shown to estimate relative quality levels reasonably accurately in a context with fixed content and artifacts typically caused by video compression and transmission errors [12,13].

It would be desirable to test the objective metrics using several independently created databases. Sequences used for training the metric should be taken with special caution and excluded from validation tests. At the time of writing, we are aware of two comprehensive lists of existing annotated video quality databases. One is the database list by Stefan Winkler on his website [14], and another is maintained by the Qualinet COST action [15]. Even though several databases are available, most of them have relatively limited scope of use. The most relevant databases, compiled from [14,15], are listed in Table I. In fact, some of the listed databases comprise two or more databases from the same institute; due to the similarities in source content and methodologies, as well as lack of space, we have not listed them as separate items. We focus on general video, and therefore we have not included the more specialized databases, such as those with 3D content.

During the past years, VQEG FR-TV Phase I database from 2000 has been used in several studies. It has been created by following the standardized test procedures rigorously, and the subjective experiments have been repeated by several institutes independently. However, the database is considered becoming outdated, since the content of interlaced NTSC and PAL resolution TV with focus on MPEG-2 compression artifacts is largely irrelevant in today's digital television broadcasting networks and internet media applications.

The recently published VQEG HDTV database from 2010 is with no doubt the largest and most significant of the publicly available video quality databases [10]. Six experiments have been conducted, each with 168 video samples containing different levels of compression and channel distortions, intended to represent realistic artifacts for an HDTV delivery system. Since the database has not been publicly available for long, research studies using the database has not yet been published at the time of writing. However, it is supposed that the database will be popular in the future, even though the full HD resolution and large size of the database may make it less attractive alternative for rapid experimentation with limited resources.

LIVE database from 2010 is probably the most popular of the databases published after VQEG FR-TV Phase I. It uses ten different source sequences in 768x432 resolution, that is a reasonable compromise between full HD and CIF resolution (352x288). Also different source distortions (MPEG-2 and H.264) and channel artifacts (IP and wireless packet loss) are covered. The weakness of the LIVE database is that it has not been cross-validated by independent test laboratories, and this is why the subjective results need to be taken with some caution.

Another recently published popular database is the EPFL-PoliMi database. Since the subjective experiment has been conducted by two laboratories independently, it can be assumed that the subjective results are reasonably reliable. The major weakness of this database is its limitations to packet loss artifacts. The test subjects have been instructed to focus on channel errors [9], and this is why the impact of coding distortion is not visible in the subjective scores. Therefore, the database is only useful for evaluating how

Table I. Summary of different subjective video quality databases available for public. Resolutions: QCIF=176x144, QVGA=320x240, CIF=352x288, 4CIF=704x576, HD720=1280x720, HD1080=1920x1080. SD (Standard Definition) covers several resolutions, approximately same as PAL resolution (768x576).

Database	Year	Resolution	Distortion types	Remarks
VQEG FR-TV Phase I	2000	SD (NTCS, PAL)	MPEG-2, packet loss	Large database, but outdated
IRCCyN/IVC	2008-09	HD, SD	H.264, packet loss	Focus on artificial loss patterns used with eye-tracking data
Poly@NYU	2008-10	CIF, QCIF, QVGA	H.264, SVC, packet loss	Focus on scalable coding
EPFL-PoliMi	2009	CIF, 4CIF	H.264, packet loss	Focus on packet loss artifacts, popular
LIVE	2009-10	SD	MPEG-2, H.264, packet loss	Different distortion types, popular
MMSPG (scalable)	2010	HD720	SVC, wavelet coding	Focus on scalable coding
VQEG HDTV	2010	HD1080	MPEG-2, H.264, packet loss	Very large database
IT-IST	2011	CIF	MPEG-2, H.264	Compression artifacts only

well the studied objective metric works with channel artifacts.

The remaining databases are less relevant for general purposes, either because of the limited number of test cases, or the limited scope. These databases may still be useful for more specific studies, but the user should be aware of their limitations. Poly@NYU and MMSPG Scalable databases focus on scalability, and they are mostly useful when resolution and frame rate are of main concern only. IRCCyN/IVC database is dominated by channel artifacts located either outside or inside the attended regions, which is a rather artificial scenario. IT-IST database does not cover channel distortion, and the sequences are in CIF resolution only.

Taking the different nature of the available databases into account, it is necessary to clarify what is the target application and distortion types the evaluated objective metric is supposed to deal with. In some cases, it may be sufficient if the metric gives reliable comparative results in the limited context, such as with fixed content or distortion type. If this is the case, it may be justified to divide the dataset into subsets (based on the source content or distortion type, for example), and perform evaluation for those subsets separately.

As a practical example, we have performed correlation analysis between subjective quality scores and the objective results obtained from PSNR, VQM and MOVIE metrics, using the LIVE and EPFL-PoliMi databases. We have observed that although PSNR performs inferior to other metrics with the whole dataset, the performance gap to VQM and MOVIE is essentially reduced, when the correlation is measured for each content separately. For some contents, PSNR performs even better than the other metrics. The result suggests that even though PSNR values are not fully comparable across different contents, PSNR gives reasonably accurate relative results in scenarios with fixed content. Of course, the result is only valid for the distortion types covered in the data sets. The explanation is most likely related to typical compression artifacts: transform coding affects in particular fine textures and sharp edges, but on the other hand, masking effect makes distortions in the detailed areas subjectively less visible than in smoother areas. This is why compressed images with a lot of fine details tend to show lower PSNR than images with large smooth surfaces, even if the subjective quality is similar. More detailed results of this study will be published elsewhere [16].

4. SELECTION OF MAPPING FUNCTION

Since practical objective quality indices and subjective scores typically have different ranges, objective video quality *VQ* needs to be mapped into predicted subjective score *MOS$_p$*, using an appropriate mapping function. In an ideal case, both objective and subjective scales were uniform, i.e. equal perceived quality difference corresponds to equal numerical difference over the whole range. In this case, linear mapping function (1) could be used.

$$MOS_p = a + b \cdot VQ \tag{1}$$

The parameters a and b are obtained by applying linear fit between objective VQ values and the respective subjective *MOS* scores. Then, *MOS$_p$* values can be solved, and correlation analysis can be performed between the predicted and actual scores, in order to evaluate how good the objective metric is.

Unfortunately, the practical objective quality scores are rarely scaled uniformly, and linear mapping may give too pessimistic view of the performance. This is why nonlinear mapping functions are widely used [4-6]. Apparently, the mapping function needs to be monotonic, at least in the relevant range of values. In the literature, the most commonly used mapping functions are cubic function (2) or logistic function (3); exponential (4) and power functions (5) may also be used [4-6,10].

$$MOS_p = a + b \cdot VQ + c \cdot VQ^2 + d \cdot VQ^3 \tag{2}$$

$$MOS_p = a / \big(1 + \exp\big(-b \cdot (VQ - c)\big)\big) \tag{3}$$

$$MOS_p = a \cdot \exp(b \cdot VQ) + c \cdot \exp(d \cdot VQ) \quad (4)$$

$$MOS_p = a \cdot VQ^b + c \quad (5)$$

In the following, we have computed the SROCC, PCC, RMSE and outlier ratios for PSNR, VQM and MOVIE metrics with EPFL-PoliMi database, using different mapping functions. We have used the objective results from our earlier study [5], and the MOS results provided in the EPFL-PoliMi database [9]. In all cases, the resulting mapping function is monotonic. This is why the order is maintained and SROCC is the same with all mapping functions. The results are summarized in Table II.

Table II. Correlation criteria when different mapping functions are used.

Mapping	Criteria	PSNR	VQM	MOVIE
All	SROCC	0.80	0.95	0.92
Linear (1)	PCC	0.79	0.90	0.82
	RMSE	0.85	0.60	0.80
	outlier	0.39	0.31	0.35
Cubic (2)	PCC	0.79	0.96	0.93
	RMSE	0.84	0.40	0.52
	outlier	0.38	0.17	0.24
Logistic (3)	PCC	0.80	0.96	0.92
	RMSE	0.84	0.40	0.55
	outlier	0.37	0.16	0.25
Power (4)	PCC	0.79	0.96	0.91
	RMSE	0.85	0.41	0.57
	outlier	0.39	0.14	0.25
Expon. (5)	PCC	0.77	0.96	0.93
	RMSE	0.89	0.39	0.51
	outlier	0.39	0.16	0.24

As seen from the results in Table II, the metrics show almost the same relative performance in respect to each other, regardless the mapping function. However, nonlinear mapping typically yields significantly higher correlations than linear mapping. VQM provides the best match with the subjective data in all cases, according to all criteria. There is no big difference between the results obtained using different nonlinear mappings. PSNR shows about the same performance even with linear mapping; this may be partially because PSNR does not take into account the characteristics of the human visual system (HVS), resulting in worse performance than the other two metrics. MOVIE is the most sensitive to different mappings: with linear mapping, it shows only slightly better performance than PSNR.

We have performed a similar analysis with some metrics also using LIVE database with results in line with the observation that the selection between different nonlinear mapping functions is not crucial, assuming that the requirement for monotonicity is fulfilled. Linear mapping may give too pessimistic view of the performance for some metrics, but on the other hand, linearity could also be considered as an additional performance indicator.

5. PERFORMANCE INDICATORS AND ANALYSIS

The purpose of correlation analysis is to study how well the analyzed objective quality metric predicts the subjective opinions on quality, expressed in MOS or DMOS. In most practical applications, the raw quality index values are less important than the capability of comparing the quality of distorted video sequences successfully. This is the case when different video compression methods working at the same bitrate are compared against each other, for example. For this purpose, SROCC is the most relevant measure, since it measures the *monotonicity* between the predicted and actual *MOS* values.

Nevertheless, it should be noted that SROCC does not always give a realistic idea of the capabilities of a metric. In particular, SROCC may underestimate performance if the metric is not capable of distinguishing subtle differences in sequences, and the data set consists of clusters with sequences having rather similar characteristics. In this case, indicators measuring the *accuracy* between predicted and actual *MOS* values, such as PCC and MSE, may define the performance of the metric more realistically.

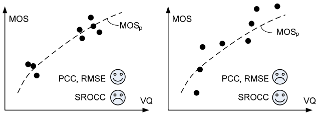


Figure 2. An example of different data sets and correlation indicators.

Figure 2. shows two artificial scenarios, where SROCC and PCC show very different performance for two data sets. In the first case, the data set contains highly clustered data points, and the rank order within those clusters is almost random. This is why SROCC is low. On the other hand, if the clusters are tight enough, PCC and RMSE may show reasonably good performance. In the second case, *VQ* predicts the rank order of the data points accurately, but the data points are arranged in an irregular shape, that cannot be modeled accurately with the standard mapping functions. This is why SROCC indicates good performance, but PCC and RMSE give less satisfactory results.

The fourth commonly used performance indicator is the outlier ratio, measuring the *consistency* of the metric. By definition in [6], outlier ratio is the proportion of MOS_p values laying outside ±2σ from *MOS*, where σ is the standard deviation of the scores obtained from the subjective test. If Gaussian distribution is assumed, ±2σ is approximately the same as 95% confidence interval. Paradoxically, outlier ratio often tells more about the accuracy of the subjective data than the performance of the objective metric: low outlier ratio does not necessarily indicate good match between the objective and subjective data, but rather a large standard deviation and therefore inaccurate subjective scores.

However, it may still be useful to include outlier ratio in the analysis, since it can reveal some cases when the metric in concern fails in spite of good performance in general. For example, a metric can be overly sensitive to certain special contents or distortion types. In case of a large σ, it may be worth consideration to compute the outlier ratio with a smaller confidence interval, for example $\pm 1.5\sigma$ or $\pm\sigma$.

According to our practical experience (see Table II, for example) and related studies [4-6], there is usually a strong correlation between PCC and RMSE results, but not always between SROCC and those other indicators. Inconsistencies are often related to weak overall results or integrated data sets containing data from different databases. Nevertheless, we believe that it is necessary to analyze correlation with different indicators, in order to reveal deficiencies in the experiments, such as inconsistent performance with different contents, distortion types or quality levels. In some cases, these inconsistencies may be related to the subjective data, rather than the performance of the objective metric. It would be highly recommended to study the reasons for anomalies via analyzing the data by hand, since this type of analysis may reveal important strengths and weaknesses either in the studied objective metric, the subjective data, or both.

6. CONCLUSIONS

In this paper, we have provided an extensive overview on the practical aspects in comparison and validation studies on objective video quality metrics. We have explained the typical procedure of evaluating objective quality metrics by analyzing the correlation between the scores obtained from subjective tests and the predicted scores produced by the objective metric in question. We have also discussed the potential problems and sources of uncertainty, related to subjective testing procedures, selection of test data, as well as processing and analyzing the data.

We have studied the reliability of subjective scores by meta-analysis and a small scale validation study based on pairwise comparisons. Our main observation is that subjective scores are reasonably consistent when similar types of distortions and contents are concerned, but subjective results across several different distortion types and content categories are more prone to inconsistencies and inaccuracies. In practice, conventional analysis based on established correlation indicators may be insufficient to reveal to what extent the results are influenced by factors related to the subjective data. This is why we would recommend researchers in the field to analyze the potential causes behind the raw numbers more carefully. In this way, the strengths and weaknesses of different subjective testing procedures and objective metrics can be revealed more comprehensively.

8. REFERENCES

[1] J. You, U. Reiter, M. M. Hannuksela, M. Gabbouj, and A. Perkis, "Perceptual-based objective quality metrics for audio-visual services - A survey," *Signal Proc. Image Communication*, vol. 25, no. 7, pp. 482-501, Aug. 2010.

[2] M. Pinson, and S. Wolf, "An Objective Method for Combining Multiple Subjective Data Sets," *Proc. of SPIE VCIP'03*, vol. 5150, Lugano, Switzerland, 2003.

[3] M. Pinson, and S. Wolf, "Comparing Subjective Video Quality Testing Methodologies," *Proc. of SPIE VCIP'03*, vol. 5150, Lugano, Switzerland, 2003.

[4] K. Seshadrinathan, R. Soundararajan, A. C. Bovik and L. K. Cormack, "Study of Subjective and Objective Quality Assessment of Video," *IEEE Trans. Image Proc.*, vol. 19, no. 6, pp.1427-1441, Jun. 2010.

[5] J. You, J. Korhonen, A. Perkis, and T. Ebrahimi, "Balancing Attended and Global Stimuli in Perceived Video Quality Assessment," *IEEE Trans. Multimedia*, vol. 13, no.6, pp.1269-1285, Dec. 2011.

[6] ITU-T SG9 Contribution 80, "Final Report from the Video Quality Experts Group on the Validation of Objective Models of Video Quality Assessment," June 2000.

[7] ITU-R Recommendation BT.500-11, "Methodology for the Subjective Assessment of the Quality of Television Pictures," International Telecommunication Union, Geneva, Switzerland, 2002.

[8] ITU-T Recommendation P.910, "Subjective Video Quality Assessment Methods for Multimedia Applications," International Telecommunication Union, 1999.

[9] F. De Simone, M. Naccari, M. Tagliasacchi, F. Dufaux, S. Tubaro, and T. Ebrahimi, "Subjective Quality Assessment of H.264/AVC Video Streaming with Packet Losses," *EURASIP J. on Image and Video Proc.*, 12 pages, 2011.

[10] Video Quality Experts Group, "VQEG Final Report of HDTV Validation Test," Jun. 2010. Available at <http:\\www.vqeg.org>.

[11] K. Seshadrinathan, R. Soundararajan, A. C. Bovik and L. K. Cormack, "A Subjective Study to Evaluate Video Quality Assessment Algorithms," *SPIE Proc. Human Vision and Electronic Imaging*, Jan. 2010.

[12] Q. Hyunh-Thu, and M. Ghanbari, "Scope of Validity of PSNR in Image/Video Quality Assessment," *Electronics Letters*, vol. 44, no. 13, pp. 800-801, Jun. 2008.

[13] U. Reiter, J. Korhonen, and J. You, "Comparing apples and oranges: Assessment of the relative video quality in the presence of different types of distortions," *EURASIP J. on Image and Video Proc.* 2011:8, Sep. 2011.

[14] S. Winkler, <http://stefan.winkler.net/resources.html>.

[15] K. Fliegel, "QUALINET Multimedia Databases v2.0," Qualinet internal document Qi0117. Sep. 2011.

[16] J. Korhonen, and J. You, "Peak Signal-to-Noise Ratio Revisited - Is Simple Beautiful?" *QoMEX'12*, Melbourne, Australia, Jul. 2012. Accepted for publication.

A.14 Quality Assessment of Images Displayed on LCD Screen with Local Backlight Dimming

C. Mantel, N. Burini, **E. Nadernejad**, J. Korhonen, and S. Forchhammer, “Quality Assessment of Images Displayed on LCD Screen with Local Backlight Dimming”, in *The Fifth International Workshop on Quality of Multimedia Experience (QoMEX)*, Jul. 2013

Reference: [14]

QUALITY ASSESSMENT OF IMAGES DISPLAYED ON LCD SCREEN WITH LOCAL BACKLIGHT DIMMING

Claire Mantel, Nino Burini, Jari Korhonen, Ehsan Nadernejad and Søren Forchhammer

Technical University of Denmark

ABSTRACT

This paper presents a subjective experiment collecting quality assessment of images displayed on a LCD with local backlight dimming using two methodologies: absolute category ratings and paired-comparison. Some well-known objective quality metrics are then applied to the stimuli and their respective performance are analyzed. The HDR-VDP metric seems to achieve good performance on every source image.

***Index Terms*—** Quality assessment, Objective metric, Local backlight dimming

1. INTRODUCTION

Liquid Crystal Displays (LCD) consists of a grid of liquid crystal (LC) cells located in front a backlight composed of light emitting diodes (LEDs), of which light is spread by a diffuser plate. The rendered image is the product of the light emitted by the backlight with the transmittance of the LC cells. Local backlight dimming technology allows controlling the LEDs composing the backlight independently from one another, as opposed to the spatially uniform illumination of conventional backlights. Local backlight dimming creates mainly two types of defects. Leakage artifact comes from the LC limitation: even at its lowest transmittance, a LC cell cannot completely block the backlight. Inversely clipping artifact appears when the backlight is not bright enough and the rendered brightness of a pixel is lower than it should be.
Evaluating the quality of images displayed on such a screen presents two issues: firstly the image displayed is not directly the signal sent to the screen but it must be modeled and secondly there is, to the best of our knowledge, no existing metric designed specifically for backlight dimming artifacts. This paper presents a subjective experiment assessing the quality of images on a LCD display with local backlight dimming and evaluates the performance of some common image quality metrics on this data using a model of the screen to simulate the rendered image.

2. EXPERIMENT DESCRIPTION

The experiment was organized in two phases using two different methodologies: paired-comparison (PC) and absolute category rating (ACR) [1]. The PC test aimed at providing precise subjective assessments of all versions for each image separately and the ACR test aimed at putting all PC results on a common scale.

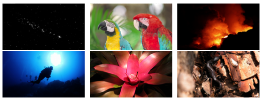


Fig. 1: Images dataset: *Stars, Parrot, Volcano, Diver, Exotic Flower* and *Lizard*. The seventh, *Beach*, can be seen in [2]

Practical setting

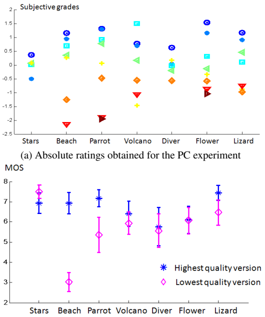
16 observers participated in the 1st phase and 15 others in the 2nd. All were naive regarding the aim of the experiment and non-experts in image quality. Both tests were performed twice and the complete duration for each participant was respectively around 30 and 45 minutes. Images were displayed on a 1920x1080 screen with a local backlight dimming of 2202 LEDs from SIM2. Observers were located at 1,76m from the screen (3 times the height of the display), at a 0 degree vision angle. The experiment was done with lights off and curtains drawn. In the PC test, observers could navigate between versions of an image for as long as they wished before choosing the one they preferred. In the ACR test, observers rated the quality of one image at a time on a 9-points scale on which only the 1 and 9 grades are qualified (resp. *very bad* and *excellent*).

Stimuli

Seven images (Fig. 1), at 1920x1080 resolution and available in perfect quality, were used for the experiment. For the 1st part, 6 backlight dimming algorithms were applied to those 7 images: Albrecht, Cho, Kang, Zhang, Nadernejad (all described in [2]) and the Gradient-Descent at power consumptions similar to the other 5 algorithms [3]. As the purpose of the second phase was to put the PC subjective results for each image on a common scale, only the preferred and least preferred versions of each image were then rated.

Data Analysis

For each image, the PC data were transformed into absolute ratings using the Thurstone-Mosteller model [4]. Figure 2a



(a) Absolute ratings obtained for the PC experiment

(b) MOS and confidence interval for the ACR experiment

Fig. 2: Subjective results

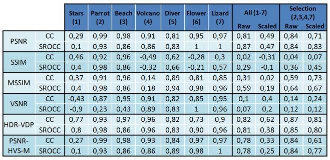
shows the obtained ratings: the higher the subjective grade is, the more participants preferred this version. ACR data was processed according to [1]: no outlier was detected and the MOS and confidence interval for each sequence are visible in Fig. 2b. For some images, the ACR and PC results are not similar (*Diver, Flower* and *Volcano* to a lesser extent) and for *Stars* they are even contradictory. One plausible explanation for this shift is that in the absence of a comparison element some backlight dimming defects are less or not noticeable. Since this tendency is even more marked for highly contrasted images it could be related to the adaptation to brightness of the Human Visual System.

The model used to simulate the displayed image is presented in [5]. The quality metrics used are: PSNR, SSIM, MSSIM, VSNR, HDR-VDP and PSNR-HVS-M [6, 7]. The performance of each metric is evaluated in Fig. 3.

3. RESULTS ANALYSIS AND PERSPECTIVES

As they have high contrast and large black areas, the images $Stars$ and $Volcano$ are those on which leakage is most visible. On the contrary the two brightest images, $Parrot$ and $Beach$, don't contain any pixel dark enough to present leakage. Globally speaking, the metrics have difficulty for *Stars* and *Volcano*, whereas they all perform quite well on $Parrot$ and $Beach$. Therefore it seems that, except for HDR-VDP, the tested metrics assess more easily clipping than leakage.

As results from the two experiments are sometimes not in line, it is delicate to merge them in a single analysis. Consequently, we think it safer to limit our analysis on the subset of the images for which the two methodologies give coherent results (*Parrot, Beach, Volcano* and *Lizard*, presented in the *Selection* column in figure 3). For this selection of images, SSIM and VSNR obtain low correlations while HDR-VDP performs best. Those results indicate that the HDR-VDP metric can efficiently assess the quality of images impaired by both clipping and leakage artifacts. Other existing metrics (PSNR, MSSIM and PSNR-HVS-M) seem to perform best on images presenting only clipping. Experiments on a larger set of images and a thorough test of the relative efficiency of PC and ACR methodologies for backlight dimming defects are needed to confirm and extend those results.



		Stars (1)	Parrot (2)	Beach (3)	Volcano (4)	Diver (5)	Flower (6)	Lizard (7)	All (1-7)		Selection (2,3,4,7)	
									Raw	Scaled	Raw	Scaled
PSNR	CC	0,29	0,99	0,98	0,91	0,81	0,95	0,97	0,81	0,49	0,84	0,71
	SROCC	0,1	0,93	0,86	0,86	0,83	1	1	0,87	0,47	0,84	0,83
SSIM	CC	0,46	0,92	0,96	-0,49	0,62	-0,28	0,3	0,02	-0,31	0,04	0,07
	SROCC	0,4	0,98	0,86	-0,32	0,66	-0,21	0,57	0,29	-0,1	0,36	0,45
MSSIM	CC	0,37	0,91	0,96	0,14	0,89	0,81	0,85	0,31	0,02	0,59	0,73
	SROCC	0,4	0,98	0,86	0,18	0,94	0,98	0,96	0,59	0,19	0,64	0,67
VSNR	CC	-0,43	0,87	0,95	0,91	0,82	0,85	0,95	0,1	0,4	0,14	0,24
	SROCC	-0,9	0,23	0,43	0,89	0,83	1	0,96	0,07	0,2	0,12	0,12
HDR-VDP	CC	0,77	0,93	0,97	0,96	0,82	0,73	0,9	0,82	0,62	0,87	0,81
	SROCC	0,8	0,98	0,86	0,96	0,83	0,90	0,96	0,81	0,38	0,85	0,80
PSNR-HVS-M	CC	0,27	0,99	0,98	0,93	0,84	0,97	0,97	0,78	0,33	0,84	0,61
	SROCC	0,1	0,93	0,86	0,86	0,89	0,98	1	0,78	0,25	0,84	0,77

Fig. 3: Pearson's correlation coefficient (CC) and Spearman's rank-order correlation (SROCC). Correlation for 'All' and 'Selection' columns are done both directly on PC absolute grades ('Raw') and by first scaling them using ACR data

4. REFERENCES

[1] ITU-R, *Recommandation BT.500-12 Methodology for the subjective assessment of the quality of television pictures*, 2009.

[2] E. Nadernejad et al., "Adaptive local backlight dimming algorithm based on local histogram and image characteristics," *SPIE 8652, Color Imaging XVIII*, 2013.

[3] N. Burini et al., "Modeling power-constrained optimal backlight dimming for color displays," *Accepted to J. Display Technol.*, 2013.

[4] J. C. Handley, "Comparative analysis of bradley-terry and thurstone-mosteller paired comparison models for image quality assessment," 2001.

[5] J. Korhonen et al., "Modeling LCD displays with local backlight dimming for image quality assessment," *Proc of the SPIE: Color Imaging XVI*, 2011.

[6] N. Ponomarenko et al., "Tid2008 - a database for evaluation of full-reference visual quality assessment metrics," *Advances of Modern Radioelectronics*, 2009.

[7] R. Mantiuk et al., "Hdr-vdp-2: a calibrated visual metric for visibility and quality predictions in all luminance conditions," *ACM Trans. Graph.*, 2011.

Appendix B

Testsets

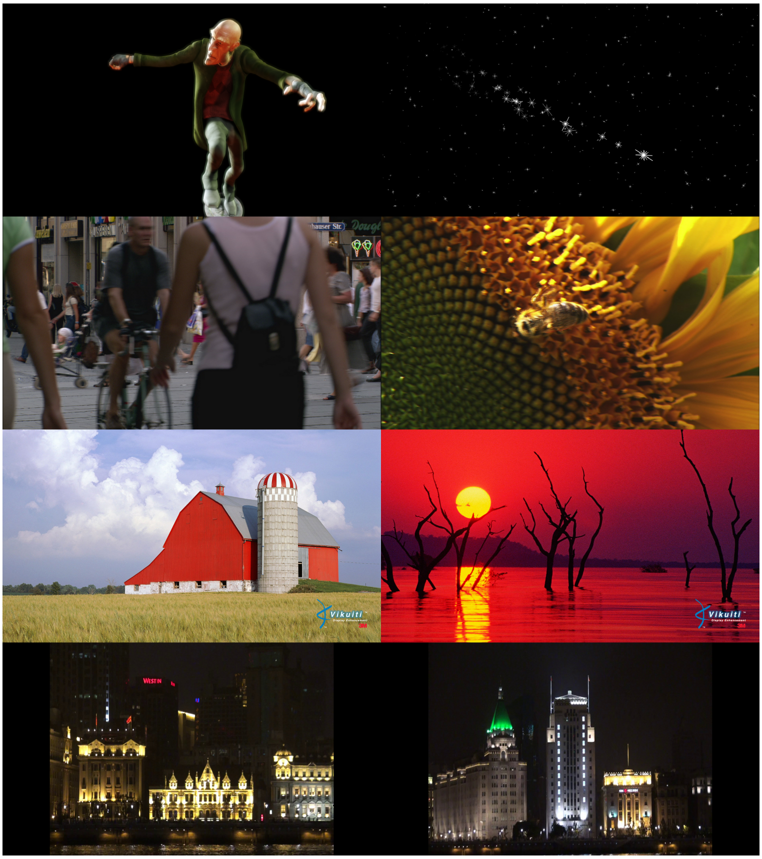


Figure B.1: This testset is called "ICIP" because it was originally used for the experiments whose results were submitted to IEEE ICIP 2012; its images are called (left to right, top to bottom) *Man*, *Pedestrian*, *City1*, *Sunflower*, *Stars*, *Barn*, *Sunset* and *City2*.

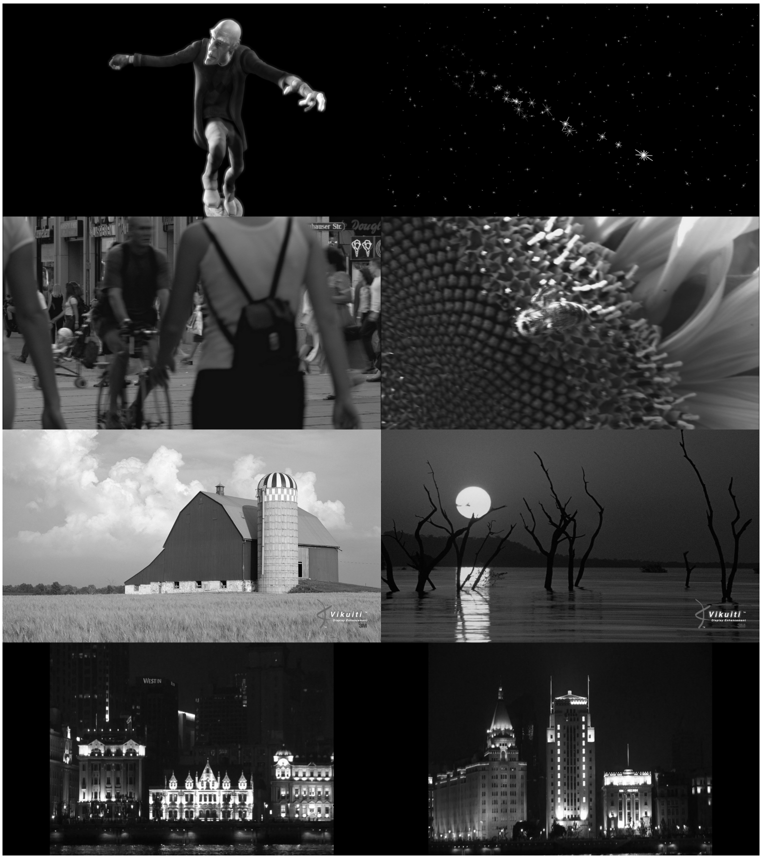


Figure B.2: This testset is called “ICIP” because it was originally used for the experiments whose results were submitted to IEEE ICIP 2012; its images are called (left to right, top to bottom) *Man*, *Pedestrian*, *City1*, *Sunflower*, *Stars*, *Barn*, *Sunset* and *City2*. Grayscale version.

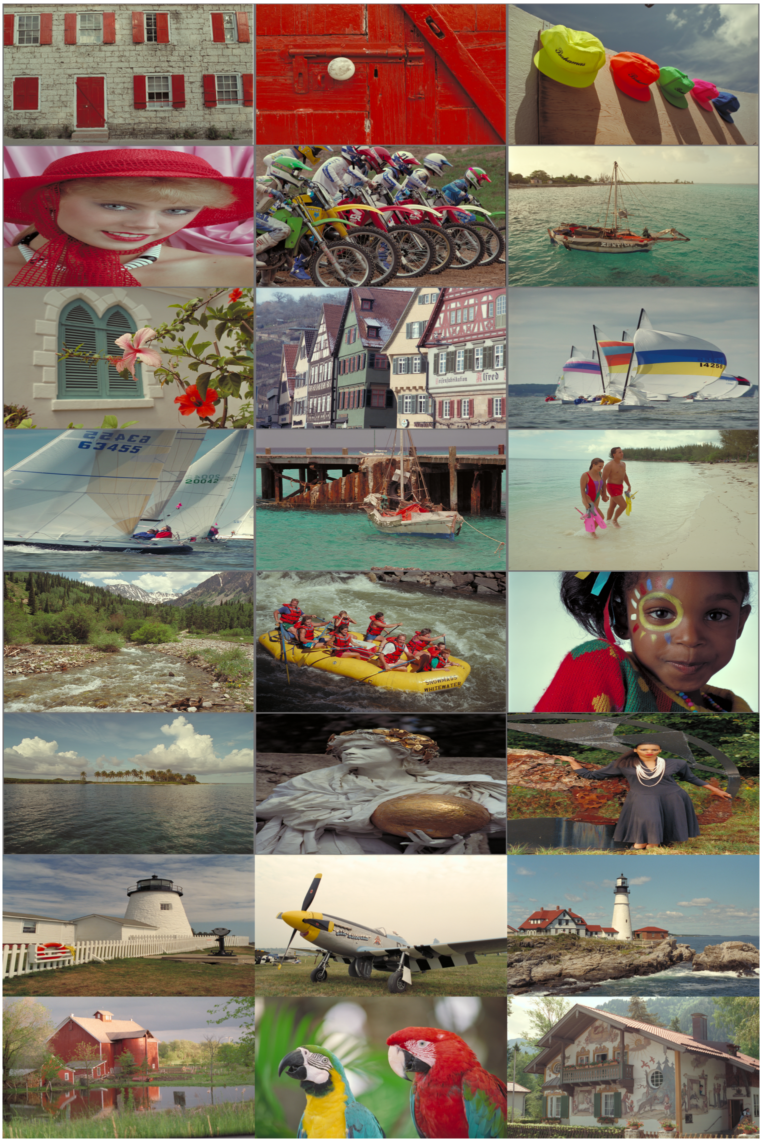


Figure B.3: Kodak True Color testset [99]. The images are called *k01*, *k02*, *k03*, *k04*, . . . , *k23*, *k24*, starting from the top-left then going left to right and top to bottom.

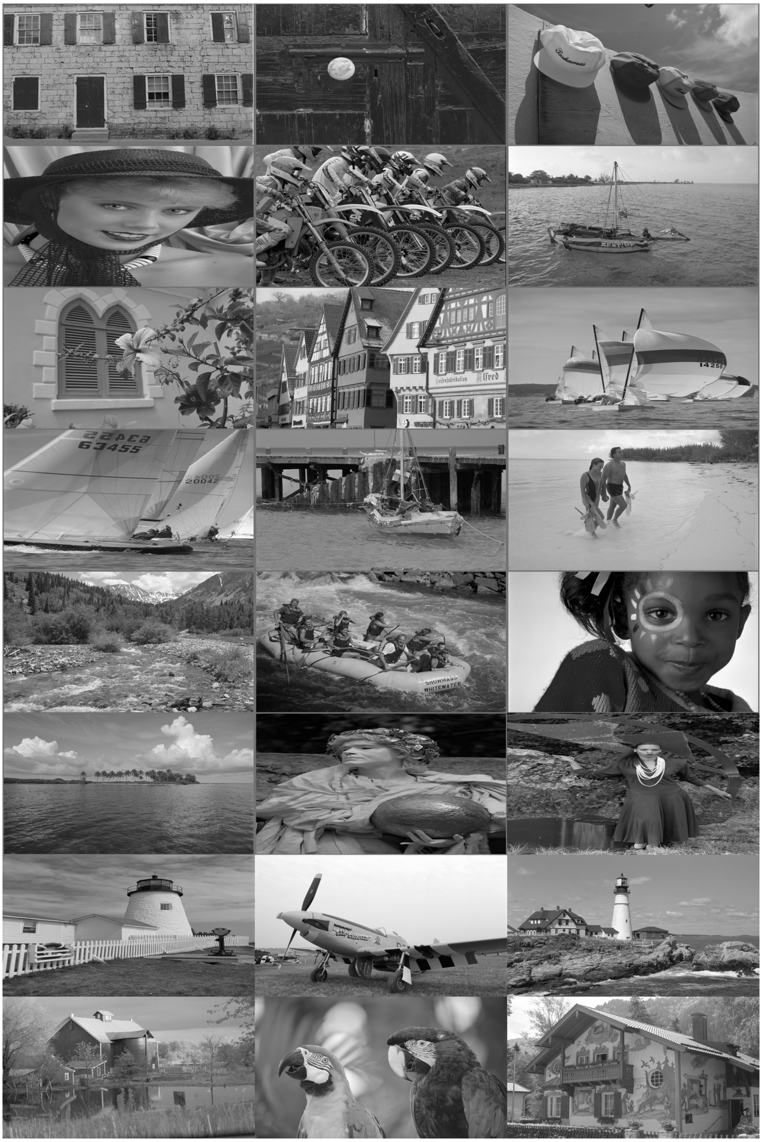


Figure B.4: Kodak True Color testset [99]. The images are called *k01*, *k02*, *k03*, *k04*, . . . , *k23*, *k24*, starting from the top-left then going left to right and top to bottom. Grayscale version.

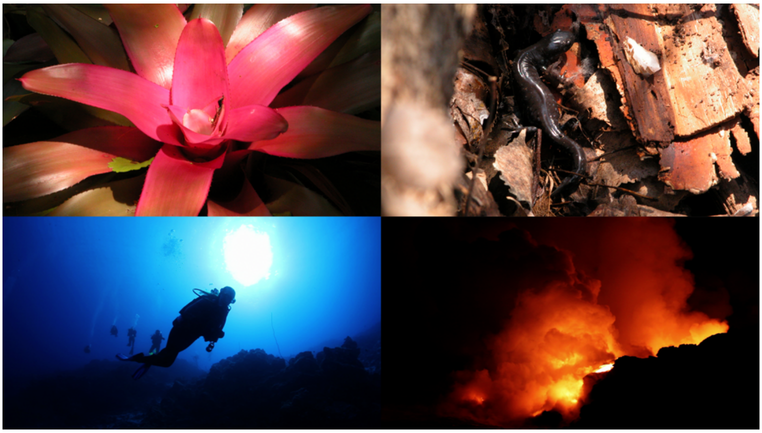


Figure B.5: Four images used for experiments: *Exotic flower*, *Lizard*, *Diver*, *Volcano*. The first two are found in [100], the other two in **cdvl**

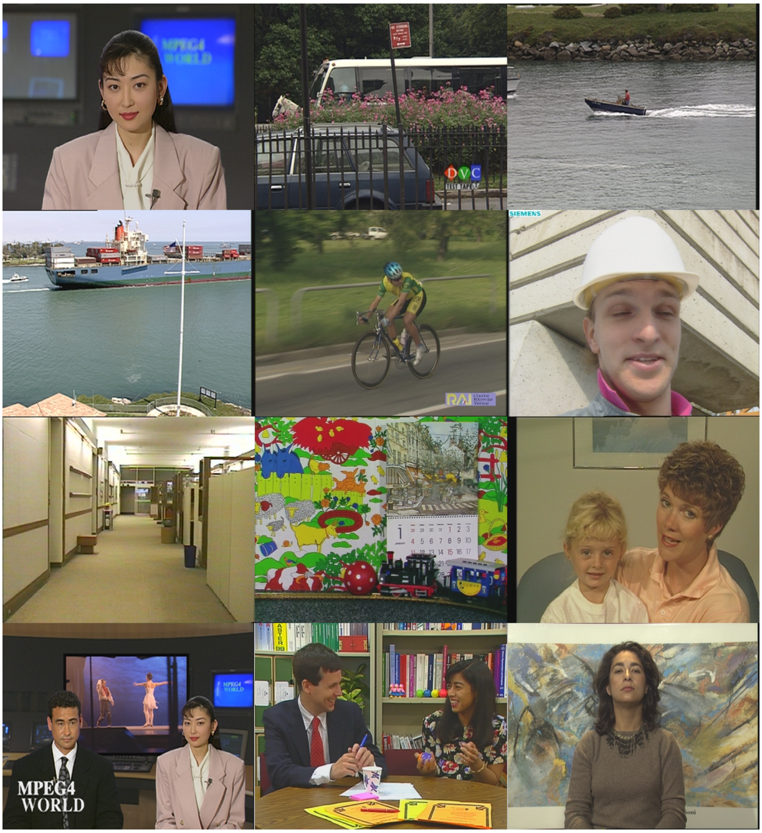


Figure B.6: First frames of 9 video sequences used for experiments: *Akiyo*, *Bus*, *Coastgard*, *Container*, *Cycling*, *Foreman*, *Hall*, *Mobile*, *Mother*, *News*, *Paris* and *Silent*.

References

[1] **E. Nadernejad**, S. Forchhammer, and J. Korhonen, “Artifact Reduction of Compressed Images and Video Combining Adaptive Fuzzy Filtering and Directional Anisotropic Diffusion”, in *2011 3rd European Workshop on Visual Information Processing (EUVIP)*, IEEE, 2011, pp. 24–29.

[2] **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.

[3] **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)*, May 2013, pp. 2508–2512.

[4] **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.

[5] **E. Nadernejad**, C. Mantel, N. Burini, and S. Forchhammer, “Flicker Reduction in LED-LCDs with Local Backlight”, in *2013 IEEE 15th International Workshop on Multimedia Signal Processing (MMSP)*, IEEE, 2013.

[6] **E. Nadernejad** and S. Forchhammer, “Wavelet-based Image Enhancement Using Fourth Order PDE”, in *2011 IEEE 7th International Symposium on Intelligent Signal Processing (WISP)*, IEEE, 2011, pp. 116–121.

[7] **E. Nadernejad**, S. Forchhammer, and S. Sharifzadeh, “Using Anisotropic Diffusion Equations in Pixon Domain for Image Denoising”, *Signal, Image and Video Processing*, 2012, ISSN: 1863-1703.

[8] **E. Nadernejad**, S. Sharifzadeh, and J. Korhonen, “Improving Performance of Wavelet-based Image Denoising Algorithm Using Complex Diffusion Process”, *Imaging Science Journal*, vol. 60, no. 4, pp. 208–218, 2012.

[9] **E. Nadernejad**, “Improvement of Nonlinear Diffusion Equation Using Relaxed Geometric Mean Filter for Low PSNR Images”, *Electronics Letters*, vol. 49, no. 7, pp. 457–458, 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*, vol. 9, no. 8, pp. 656–665, 2013.

[11] C. Mantel, N. Burini, **E. Nadernejad**, J. Korhonen, S. Forchhammer, and J. Pedersen, “Controlling Power Consumption for Displays With Backlight Dimming”, *Display Technology, Journal of*, vol. 9, no. 12, pp. 933–941, 2013.

[12] N. Burini, C. Mantel, **E. Nadernejad**, J. Korhonen, S. Forchhammer, and J. Pedersen, “Block-Based Gradient Descent for Local Backlight Dimming and Flicker Reduction”, *IEEE/OSA Journal of Display Technology*, vol. PP, no. 99, pp. 1–1, 2013. DOI: 10.1109/JDT.2013.2286213.

[13] 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.

[14] C. Mantel, N. Burini, **E. Nadernejad**, J. Korhonen, and S. Forchhammer, “Quality Assessment of Images Displayed on LCD Screen with Local Backlight Dimming”, in *The Fifth International Workshop on Quality of Multimedia Experience (QoMEX)*, Jul. 2013.

[15] 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.

[16] ——, “Image Dependent Energy-Constrained Local Backlight Dimming”, in *IEEE International Conference on Image Processing (ICIP)*, Sep. 2012, pp. 2797–2800.

[17] **E. Nadernejad** and M. Nikpour, “Image Denoising Using New Pixon Representation Based on Fuzzy Filtering and Partial Differential Equations”, *Digital Signal Processing*, vol. 22, no. 6, pp. 913–922, 2012.

[18] **E. Nadernejad** and A. Barari, “A Novel Pixon-Based Image Segmentation Process Using Fuzzy Filtering and Fuzzy C-mean Algorithm”, *International Journal of Fuzzy Systems*, vol. 13, no. 4, pp. 350–357, 2011.

[19] **E. Nadernejad** and S. Sharifzadeh, “A New Method for Image Segmentation Based on Fuzzy C-means Algorithm on Pixonal Images Formed by Bilateral Filtering”, *Signal, Image and Video Processing*, vol. 7, no. 5, pp. 855–863, 2013.

[20] **E. Nadernejad**, H. Hassanpour, and H. Naimi, “High-speed MRF-based Segmentation Algorithm Using Pixonal Images”, *Imaging Science Journal*, vol. 61, no. 7, pp. 592–600, 2013.

[21] M. Salarian, **E. Nadernejad**, and H. MiarNaimi, “A New Modified Fast Fractal Image Compression Algorithm”, *Imaging Science Journal*, vol. 61, no. 2, pp. 219–231, 2013.

[22] “Progress of LED Backlights for LCDs”, *Journal of the Society for Information Display*, vol. 16, no. 2, pp. 287–310, 2008.

[23] 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.

[24] W. B. Pennebaker and J. L. Mitchell, *JPEG: Still Image Data Compression Standard*. Springer, 1992.

[25] J. Watkinson, *MPEG-2*, 2nd ed. Focal Press, 1999.

[26] I. E. Richardson, *The H.264 Advanced Video Compression Standard*, 2nd ed. Wiley, 2010.

[27] G. Zhai, W. Zhang, W. Yang Xiaokang Lin, and Y. Xu, “Efficient Deblocking with Coefficient Regularization, Shape Adaptive Filtering and Quantization Constraint”, *IEEE Transactions on Multimedia*, vol. 10, no. 5, pp. 735–745, Aug. 2008.

[28] C. Kim, “Adaptive Post-filtering for Reducing Blocking and Ringing Artifacts in Low Bit-rate Video Coding”, *Signal Processing: Image Communication*, vol. 17, no. 7, pp. 525–535, 2002.

[29] J. Ren, J. Liu, M. Li, W. Bai, and Z. Guo, “Postprocessing of Block-Coded Videos for Deflicker and Deblocking”, in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2013, pp. 2508–2512.

[30] ——, “Image Blocking Artifacts Reduction via Patch Clustering and Low-Rank Minimization”, in *Data Compression Conference (DCC)*, 2013, pp. 516–516.

[31] Y.-Y. Chen, Y.-W. Chang, and W.-C. Yen, “Design a Deblocking Filter with Three Separate Modes in DCT-based Coding”, *Journal of Visual Communication and Image Representation*, vol. 19, no. 4, pp. 231–244, 2008.

[32] C. H. Yeh, S. F. Jiang, T. Ku, M. Chen, and J. Jhu, “Postprocessing Deblocking Filter Algorithm for Various Video Decoders”, *Image Processing, IET*, vol. 6, no. 5, pp. 534–547, 2012.

[33] A. Saleem, A. Beghdadi, and B. Boashash, “Improving Performance of Deblocking Techniques Using Image Fusion”, in *Information Science, Signal Processing and their Applications (ISSPA), 2012 11th International Conference on*, 2012, pp. 389–394.

[34] N. C. Francisco, N. M. Rodrigues, E. A. da Silva, and S. M. de Faria, “A Generic Post-deblocking Dilter for Block Based Image Compression Algorithms”, *Signal Processing: Image Communication*, vol. 27, no. 9, pp. 985–997, 2012.

[35] T. Chen, H. Wu, and B. Qiu, “Adaptive Postfiltering of Transform Coefficients for the Reduction of Blocking Artifacts”, *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 11, pp. 594–602, 2001.

[36] S. Liu and A. Bovik, "Efficient DCT-domain Blind Measurement and Reduction of Blocking Artifacts", *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 12, pp. 1139–1149, 2002.

[37] Z. Xu and S. Xie, "A Deblocking Algorithm Using Anisotropic Diffusion Based on HVS", *Computer Engineering*, vol. 32, no. 4, pp. 10–12, 2006.

[38] E. Choi and M. Kang, "Deblocking Algorithm for DCT-based Compressed Images Using Anisotropic Diffusion", in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 3, 2003, pp. 717–720.

[39] S. Yao, K. P. Lim, L. Xiao, and S. Rahardja, "A Post-processing Algorithm Using Histogram-driven Anisotropic Diffusion", in *Proceedings of IEEE International Symposium on Circuits and Systems (ISCAS)*, vol. 5, 2005, pp. 4233–4236.

[40] S. Oguz, Y. Hu, and T. Nguyen, "Image Coding Ringing Artifact Reduction Using Morphological Post-filtering", in *Proceedings of IEEE International Workshop on Multimedia Signal Procrocessing (MMSP)*, 1998, pp. 628–633.

[41] Y. Nie, H.-S. Kong, A. Vetro, H. Sun, and K. Barner, "Fast Adaptive Fuzzy Post-filtering for Coding Artifacts Removal in Interlaced Video", in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2005, pp. 993–996.

[42] X. Fan, W. Gao, Y. Lu, and D. Zhao, "Flicking Reduction in All Intra Frame Coding", Joint Video Team of ISO/IEC MPEG and ITU-T VCEG, Tech. Rep. JVT-E070, Oct. 2002.

[43] A. Leontaris, Y. Tonomura, T. Nakachi, and P. Cosman, "Flicker Suppression in JPEG2000 Using Segmentation-based Adjustment of Block Truncation Lengths", in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 1, 2007, pp. 1117–1120.

[44] D. Vo, T. Nguyen, S. Yea, and A. Vetro, "Edge-based Directional Fuzzy Filter for Compression Artifact Reduction in JPEG Images", in *Proceedings of IEEE International Conference on Image Processing (ICIP)*, 2008, pp. 797–800.

[45] ——, "Adaptive Fuzzy Filtering for Artifact Reduction in Compressed Images and Videos", *IEEE Transactions on Image Processing*, vol. 18, no. 6, pp. 1166–1178, May 2009.

[46] N. Yao and E. Kenneth, "Fuzzy Weighted Median Filters", in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 2, 2002, pp. 1125–1128.

[47] S.-C. Tai, Y.-Y. Chen, and S.-F. Sheu, "Deblocking Filter for Low Bit Rate MPEG-4 Video", *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 15, no. 6, pp. 733–741, Jun. 2005.

[48] C. Chen, C. Huang, Y. Chen, S. Chien, and L. Chen, "System Analysis of VLSI Architecture for 5/3 and 1/3 Motion-compensated Temporal Filtering", *IEEE Transactions on Image Processing*, vol. 54, no. 10, pp. 4004–4014, 2006.

[49] W. Zhai G. Lin, J. Cai, X. Yang, and W. Zhang, "Efficient Quadtree Based Block-shift Filtering for Deblocking and Deringing", *Journal of Visual Communication and Image Representation*, vol. 20, no. 8, pp. 595–607, Feb. 2009.

[50] S.-H. Shin, K.-H. Park, and T.-Y. Chai Y.-J.and Kim, "Variable Block-based Deblocking Filter for H.264/AVC on Low-end and Low-bit Rates Terminals", *Signal Processing: Image Communication*, vol. 25, pp. 255–267, 2010.

[51] W. Mahjoub and G. Osman H. Aly, "H.264 Deblocking Filter Enhancement", in *International Conference on Computer Engineering & Systems*, 2011, pp. 219–224.

[52] S. Forchhammer, H. Li, and J. D. Andersen, "No-reference Analysis of Decoded MPEG Images for PSNR Estimation and Postprocessing", *Journal of Visual Communication and Image Representation*, vol. 22, no. 4, pp. 313–324, 2011.

[53] J. Daugman, "Spatial Visual Channels in the Fourier Plane", *Vision Research*, vol. 24, no. 9, pp. 891–910, 1984.

[54] P. Perona and J. Malik, “Scale-space and Edge Detection Using Anisotropic Diffusion”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 127, pp. 629–639, 1990.

[55] S. Chebbo, P. Durieux, and B. Pesquet-Popescu, “Adaptive Deblocking Filter for DCT Coded Video”, in *Proceedings of the International Workshop on Video Processing and Quality Metrics for Consumer Electronics (VPQM)*, Jan. 2009.

[56] X. Huang, H. Li, and S. Forchhammer, “A Multi-Frame Post-Processing Approach to Improved Decoding of H.264/AVC Video”, in *Proceedings of IEEE International Conference on Image Processing (ICIP)*, vol. 4, 2007, pp. 381–384.

[57] M. Zhang and B. K. Gunturk, “Multiresolution Bilateral Filtering for Image Denoising”, *IEEE Transactions on Image Processing*, vol. 17, no. 12, pp. 2324–2333, Dec. 2008.

[58] H.-S. Kong, Y. Nie, A. Vetro, H. Sun, and K. Barner, “Adaptive Fuzzy Post-filtering for Highly Compressed Video”, in *IEEE Conference on Image Processing (ICIP)*, vol. 3, Oct. 2004, pp. 1803–1806.

[59] M. Zhang and B. K. Gunturk, “Compression Artifact Reduction with Adaptive Bilateral Filtering”, in *Proceedings of the SPIE*, vol. 7257, 2009, pages.

[60] S. D. Kim, J. Yi, H. M. Kim, and J. B. Ra, “A Deblocking Filter with Two Separate Modes in Block-based Video Coding”, *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 9, no. 1, pp. 156–160, Feb. 1999.

[61] F. De Simone, D. Ticca, F. Dufaux, M. Ansorge, and T. Ebrahimi, “A Comparative Study of Color Image Compression Standards Using Perceptually Driven Quality Metrics”, in *Proceedings of SPIE Optics and Photonics*, 2008.

[62] P. Choudhury and J. Tumblin, “The Trilateral Filter for High Contrast Images and Meshes”, in *ACM SIGGRAPH 2005 Courses*, ser. SIGGRAPH ’05, Los Angeles, California: ACM, 2005.

[63] W. Wong, A. Chung, and S. Yu, “Trilateral Filtering for Biomedical Images”, in *Biomedical Imaging: Nano to Macro, 2004. IEEE International Symposium on*, vol. 1, 2004, pp. 820–823.

[64] T. Wang and G. Zhai, “JPEG2000 Image Postprocessing with Novel Trilateral Deringing Filter”, *Optical Engineering*, vol. 47, no. 2, pages, 2008.

[65] Y. Luo and R. Ward, “Removing the Blocking Artifacts of Block-based DCT Compressed Images”, *Image Processing, IEEE Transactions on*, vol. 12, no. 7, pp. 838–842, 2003.

[66] N. Burini, “Processing Decoded Video for Backlight Dimming ’Video Quality Enhancement on LCD with Dynamic Local Backlight”, PhD thesis, Technical University of Denmark, 2013.

[67] S. M. Shunsuke Kobayashi and S. Lim, *LCD Backlights.* Wiley - SID Series in Display Technology, 2009.

[68] A. A. Michelson, “Studies in Optics”, PhD thesis, University of Chicago Press, Chicago, Illinois, 1927.

[69] 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.

[70] 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.

[71] 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.

[72] 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.

[73] 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.

[74] 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.

[75] 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.

[76] 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.

[77] 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.

[78] 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.

[79] 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.

[80] 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.

[81] G.-Z. Wang, Y.-P. Huang, F.-C. Lin, and H.-P. D. Shieh, "Delta-Color Adjustment (DCA) for Spatial Modulated Color Backlight Algorithm on High Dynamic Range CD TVs", *IEEE/OSA Journal of Display Technology*, vol. 6, no. 6, pp. 215–220, Jun. 2010.

[82] S.-J. Kang and Y. H. Kim, "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.

[83] 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.

[84] 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.

[85] H. Nam, “Low Power Active Dimming Liquid Crystal Display with High Resolution Backlight”, *Electronics Letters*, vol. 47, p. 538, 9 2011.

[86] M. Grant and S. Boyd, *CVX: matlab software for disciplined convex programming, version 1.21*, `http://cvx.com/cvx`, Apr. 2011.

[87] ——, “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.

[88] 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*, vol. 8305, 2012, p. 830 517.

[89] 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.

[90] *SIM2 HDR TV*, `http://www.sim2.com/HDR/`.

[91] J. C. Handley, “Comparative Analysis of Bradley-Terry and Thurstone-Mosteller Paired Comparison Models for Image Quality Assessment”, in *Im. Proc., Im. Qual., Im. Capture Syst. Conf. (PICS’01)*, 2001, pp. 108–112.

[92] R. L. Woods, P. Satgunam, P. M. Bronstad, and E. Peli, “Statistical Analysis of Subjective Preferences for Video Enhancement”, in *Proceedings of the SPIE Human Vision and Electronic Imaging*, vol. 7527, 2010.

[93] K. Tsukida and M. R. Gupta, “How to Analyze Paired Comparison Data”, UWEE, Tech. Rep., 2011.

[94] L. Kerofsky and J. Zhou, “59.3: Temporal Filtering In LCD Backlight Modulation”, *SID Symposium Digest of Technical Papers*, vol. 39, no. 1, pp. 903–906, 2008.

[95] W.-C. Cheng, Y. Hou, and M. Pedram, “Power Minimization in a Backlit TFT-LCD Display by Concurrent Brightness and Contrast Scaling”, in *Design, Automation and Test in Europe Conference and Exhibition, 2004. Proceedings*, vol. 1, 2004, 252–257 Vol.1.

[96] J. H. Lee, S.-E. Kim, T.-H. Lee, W.-J. Song, M. K. Kim, T. W. Lee, and C. G. Kim, “P-54: Flicker Reducing Backlight Control Based on Adaptive Moving Average Filtering”, *SID Symposium Digest of Technical Papers*, vol. 40, no. 1, pp. 1294–1297, 2009.

[97] “Sita sings the blues”, [Online]. Available: `www.sitasingstheblues.com`.

[98] “Consumer digital video database”, [Online]. Available: `http://www.cdvl.org`.

[99] *Kodak True Color Image Suite*, `http://r0k.us/graphics/kodak/`.

[100] 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.

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 Photographic Experts Group

LC Liquid Crystal

LCD Liquid Crystal Display

LED Light Emitting Diode

LUT Look-Up Table

MAE Mean Absolute Error

MPEG Motion Photographic 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 Spread 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

IIR Infinite Impulse Response

MA Moving Average

ISBN: 978-87-93089-18-1

Published by:
DTU Fotonik
Department of Photonic Engineering
Technical University of Denmark
Ørstyeds Plads, building 343
DK-2800 Kgs. Lyngby

Ehsan Nadernejad was born in Babol (Iran) in 1979 and received the B.S. and M.S. degrees in electronic engineering from Mazandaran University, Iran, in 2003 and 2007, respectively. After working for some years in Iran, he move to Denmark in September 2010 and started his PhD at DTU, in the Department of Photonics Engineering (DTU Fotonik). His supervisor was Prof Søren Forchhammer. His research interests include signal, image and video processing algorithms, Quality assessment and display technology.

DTU Fotonik
Department of Photonics Engineering