Capturing and Post-Processing of Stereoscopic 3D Content for Improved Quality of Experience

by

Di Xu

B.Sc., Beijing Normal University, 2003

M.A.Sc., University of Victoria, 2007

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF

Doctor of Philosophy

in

THE FACULTY OF GRADUATE STUDIES

(Electrical & Computer Engineering)

THE UNIVERSITY OF BRITISH COLUMBIA

(Vancouver)

February 2013

Abstract

3D video can offer real-life viewing experience by providing depth impression. 3D technology has not yet been widely adopted due to challenging 3D-related issues, ranging from capturing to post-processing and display. At the capturing side, lack of guidelines may lead to artifacts that cause viewers headaches and nausea. At the display side, not having 3D content customized to a certain aspect ratio, display size, or display technology may result in reduced quality of experience. Combining 3D with high-dynamic-range imaging technology adds exciting features towards real-life experience, whereas conventional low-dynamic-range content often suffers from color saturation distortion when shown on high-dynamic-range displays. This thesis addresses three important issues on capturing and post-processing 3D content to achieve improved quality of experience.

First, we provide guidelines for capturing and displaying 3D content. We build a 3D image and video database with the content captured at various distances from the camera lenses and under different lighting conditions. We conduct comprehensive subjective tests on 3D displays of different sizes to determine the influence of these parameters to the quality of 3D images and videos before and after horizontal parallax adjustment.

Next, we propose a novel and complete pipeline for automatic content-aware 3D video reframing. We develop a bottom-up 3D visual attention model that identifies the prominent regions in a 3D video frame. We further provide a dynamic bounding box that crops the video and avoids annoying problems, such as jittering and window violation. Experimental results show that our algorithm is both effective and robust.

Finally, we propose two algorithms for correcting saturation in color images and videos. One algorithm uses a fast Bayesian-based approach that utilizes images' strong spatial correlation and the correlations between the R, G, and B color channels. The other algorithm takes advantage of the strong correlation between the chroma of the saturated pixels and their surrounding unsaturated pixels. Experimental results show that our methods effectively correct the saturated 2D and 3D images and videos. Our algorithms significantly outperform the existing state-of-the-art method in both objective and subjective qualities, resulting in plausible content that resembles real-world scenes.

Preface

This thesis presents research conducted by Di Xu, in collaboration with Drs. Lino Coria and Colin Doutre, under the guidance of Dr. Panos Nasiopoulos. A list of publications resulting from the work presented in this thesis is provided on the following pages.

The work presented in Chapter 2 has been published in [P1-P4] and submitted to [P5]. The content of Chapter 3 is submitted to [P6], a provisional patent application has been filled based on the material in the chapter [P7], part of the chapter is published in [P8], and an application is published in [P9]. Chapter 4 appears in [P10-P12].

The work presented in Chapters 2 and 3 of this thesis was performed by Di Xu and Dr. Lino Coria, including data acquisition, algorithm designing, and manuscript writing. Di Xu was the main contributor for implementing the proposed algorithms, conducting the experiments, and analyzing the results. Dr. Panos Nasiopoulos provided guidance and editorial input into the manuscript writing.

The work presented in Chapter 4 was primarily performed by Di Xu, including designing and implementing the proposed algorithms, performing all experiments, analyzing the results, and writing the manuscripts. The work was conducted with suggestions, feedback, and manuscript editing from Dr. Colin Doutre and the guidance and editorial input of Dr. Panos Nasiopoulos.

The first and last chapters of this thesis were written by Di Xu, with editing assistance from Dr. Nasiopoulos.

List of Publications Based on Work Presented in This Thesis

Chapter 2

[P1] D. Xu, L. E. Coria, and P. Nasiopoulos, "Guidelines for an Improved Quality of Experience in 3D TV and 3D Mobile Displays," *Journal of the Society for Information Display*, vol. 20, no. 7, pp. 397–407, July 2012.

[P2] D. Xu, L. E. Coria, and P. Nasiopoulos, "Quality of Experience for the Horizontal Pixel Parallax Adjustment of Stereoscopic 3D Videos," in *Proc. of IEEE International Conference on Consumer Electronics (ICCE)* , Las Vegas, NV, USA, Jan. 13-16, 2012, pp. 398-399.

[P3] L. Coria, D. Xu, and P. Nasiopoulos, "Quality of Experience of Stereoscopic Content on Displays of Different Sizes: A Comprehensive Subjective Evaluation," in *Proc. of IEEE International Conference on Consumer Electronics (ICCE)*, Las Vegas, NV, USA, Jan. 9-12, 2011, pp. 778-779.

[P4] D. Xu, L. Coria, and P. Nasiopoulos, "Guidelines for Capturing High Quality Stereoscopic Content Based on a Systematic Subjective Evaluation," in *Proc. of IEEE International Conference on Electronics, Circuits, and Systems, (ICECS)*, Athens, Greece, Dec. 12-15, 2010, pp. 166-169.

[P5] D. Xu, L. Coria, and P. Nasiopoulos, “A quality metric for 3D content: The effect of capturing parameters,” submitted, Feb. 2013.

Chapter 3

[P6] D. Xu, L. Coria, and P. Nasiopoulos, "Smart Stereoscopic 3D Video Reframing Based on a 3D Visual Attention Model," submitted, Nov. 2012.

[P7] D. Xu, L. Coria, and P. Nasiopoulos, “Automatic Stereoscopic 3D Video Reframing,” US Provisional Patent Application No. 61/714119, filed October 15, 2012.

[P8] L. Coria, D. Xu, and P. Nasiopoulos, "Automatic Stereoscopic 3D Video Reframing," in *Proc. of 3DTV-CONFERENCE 2012*, ETH Zurich, Switzerland, Oct. 15-17, 2012.

[P9] M. T. Pourazad, D. Xu, and P. Nasiopoulos, "Random Forests Based View Generation for Multiview TV," in *Proc. of IEEE International Conference on Tools with Artificial Intelligence (ICTAI)* , Athens, Greece, Nov. 7-9, 2012.

Chapter 4

[P10] D. Xu, C. Doutre, and P. Nasiopoulos, "Correction of Clipped Pixels in Color Images," *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, no. 3, pp. 333-344, Mar. 2011.

[P11] D. Xu, C. Doutre, and P. Nasiopoulos, "An Improved Bayesian Algorithm for Color Image Desaturation," in *Proc. of IEEE International Conference on Image Processing (ICIP)*, Hong Kong, Sep. 2010, pp. 1325-1328.

[P12] D. Xu, C. Doutre, and P. Nasiopoulos, "Saturated-Pixel Enhancement for Color Images," in *Proc. of IEEE International Symposium on Circuits and Systems (ISCAS)*, Paris, France, May 30th to Jun. 2nd, 2010, pp. 3377-3380.

Table of Contents

List of Tables

List of Figures

List of Acronyms

3D	Three Dimensional
CRT	Cathode Ray Tube
FIR	Finite Impulse Response
HD	High Definition
HDR	High Dynamic Range
HVS	Human Visual System
ITM	Inverse Tone Mapping
JVT	Joint Video Team
LCD	Liquid Crystal Display
LDR	Low Dynamic Range
MPEG	Moving Picture Experts Group
MOS	Mean Opinion Score
NCC	Normalized Cross-Correlation
PSNR	Peak Signal-to-Noise Ratio
QoE	Quality of experience
QT	Quaternion Transform
RGB	Red, Green, Blue (colour space)
S3D	Stereoscopic Three Dimensional
SIFT	Scale-Invariant Feature Transform
SSIM	Structural Similarity Index
TM	Tone Mapping
VAM	Visual Attention Model
VDP	Visible Difference Predictor

Acknowledgments

This thesis would never have been done without the help and support from numerous people. Needless to say, I thank all of them. In particular, I would like to take this opportunity to express my thankfulness to the following individuals.

First and foremost, I give my sincere gratitude to my supervisor Dr. Panos Nasiopoulos for his continuous guidance and support throughout my Ph.D. studies. I thank Panos for his encouragement, patience, enthusiasm, and immense knowledge in multimedia. He has always been a great mentor, a role model, and a friend. I could not imagine a better supervisor for my Ph.D. studies.

I would also like to thank my dissertation committee and chair, Dr. Jane Wang, Dr. Shahriar Mirabbasi, Dr. John Madden, Dr. Michiel van de Panne, and Dr. Vincent Wong, for investing time and energy in my studies. Their encouragement and insightful comments have been extremely valuable to me.

Many thanks also go to Dr. Lino Coria, Dr. Colin Doutre, and Dr. Mahsa Pourazad. I was lucky to have the opportunities to collaborate with these brilliant researchers. I also thank my other colleagues at the UBC Digital Multimedia Lab, Dr. Qiang Tang, Dr. Zicong Mai, Dr. Hassan Mansour, Dr. Matthias von dem Knesebeck, Dr. Victor Sanchez, Sergio Infante, Ashfiqua Connie, Mohsen Amiri, Sima Valizadeh, Maryam Azimi, Cara Dong, Amin Banitalebi, Iliya Koreshev, Basak Oztas, Bambang Sarif, and Hamid Reza Tohidypour. I thank you all for creating a vibrant and supportive research environment,

for the stimulating discussions, for your friendship, and for all the fun we have had together.

Next, I would like to express my gratitude to the following individuals, who have kindly helped me in one way or another during my Ph.D. studies: Dr. Rabab Ward, Dr. Michael Adams, Dr. Mehrdad Fatourechi, Dr. Xin Yi Yong, Dr. Angshul Majumdar, Tanaya Guha, Mani Malek, Ehsan Nezhadarya, Simon Fauvel, Ehsan Vahedi, Haoming Li, Chenjuan Zhou, Emily Yao, Amy You Wei, Chen He, Huasong Cao, Xudong Lv, Aiping Liu, Xun Chen.

I thank those who participated in my subjective tests. Without your time and conscientious attitude, this dissertation would not have been possible. I would also like to thank the Natural Sciences and Engineering Research Council of Canada (NSERC) and the University of British Columbia for providing financial support in the form of research grants and tuition awards.

Last but not least, I thank my dear parents, husband, and son for their constant love, support, and encouragement.

Dedication

To my family

1 Introduction and Overview

3D video can offer real-life viewing experience by providing depth impression. Although 3D technology has been under development for more than a century, it has not been widely adopted by the consumer market compared with the conventional 2D counterpart. This is due to many challenging 3D-related issues, ranging from capturing, compression, transmission, to displaying and content post-processing.

In recent years, Hollywood studios produce most of their high-budget movies in 3D. A common complaint from many 3D viewers, especially those watching content on home 3D displays, is the headache, nausea, and/or visual fatigue. Although viewers are often amazed by the pop-out 3D effects, extensive use of it tends to introduce heavy visual fatigue. Presently, Hollywood producers use empirical experience to ensure that they create high quality 3D content. Unfortunately, the effects of the capturing rules followed have not been quantified nor systematically proven to achieve a high quality 3D viewing experience.

Advances in 3D technology have made 3D displays and videos more popular in the recent years. A variety of 3D displays are available, ranging from high-resolution 3D TVs with a typical 16:9 aspect ratio to low-resolution mobile devices, which often have 4:3 and 3:2 aspect ratios. 3D images and videos usually have to undergo changes in size and aspect ratio to adapt to different displays. 3D content shown with a wrong aspect ratio (inconsistent with the display) suffers from distortions or lose of resolution, which significantly reduces its perceived quality.

In summary, although there has been a lot of progress in the area of 3D, 3D video systems can only be a lasting success if the perceived image quality and viewing comfort are significantly better than those of conventional 2D systems. Current 3D technologies fail to meet these criteria. 3D capturing, processing and display technologies need to be improved, and a more realistic reproduction of contrast and color is needed. The latter can be accomplished by the use of high dynamic range (HDR) imaging and display, which will provide a greater range of luminance and a wider color gamut. It is, thus, of great interest to combine the 3D immersive experience with HDR capabilities in order to produce a true to life viewing experience.

In this thesis, we propose novel techniques for capturing and post-processing stereoscopic 3D content that ensure high quality 3D viewing experience. In Chapter 2, we provide capturing guidelines for an improved viewing experience in 3D TV and 3D mobile displays. Chapter 3 presents a smart video reframing technology that automatically adjusts aspect ratios for 3D video while keeping the most important content within the cropped frames. In Chapter 4, we describe two color correction methods for enhancing the quality of HDR images and videos generated from low dynamic range (LDR) content.

The following sections of this introduction and overview chapter offer the background information on 3D and HDR technologies as well as literature reviews of the topics covered in each of the following research chapters. Section 1.1 introduces 3D and HDR related technologies, including 3D video representation, 3D displays, HDR imaging technology, and quality metrics for visual media. Section 1.2 presents camera setup

options for 3D capturing and the need for horizontal parallax adjustment for regulating depth perception. A literature review on 2D and 3D reframing is provided in Section 1.3. Section 1.4 states the background information and existing methods for color desaturation. Section 1.5 concludes the introduction with a summary of the scientific contributions of this thesis.

1.1 3D and High Dynamic Range Video Technology Overview

Stereoscopic 3D (S3D) refers to the technique that represents 3D video with two offset (left and right) videos that will be viewed separately by the left and right eyes. These two-dimensional videos are then combined in the brain to give the perception of 3D. In Sections 1.1.1 and 1.1.2, we provide an overview of the major techniques used in S3D video representation and S3D displays, respectively. An overview of HDR technology is given in Section 1.1.3, followed by a brief review of the available quality assessment methods in Section 1.1.4.

1.1.1 Stereoscopic 3D Video Representation

S3D is currently the most popular 3D format. It is widely used in 3D cinema, 3D Blu-ray discs, 3D displays, 3D projectors, and 3D broadcasting, such as cable and the Internet. There are several ways to represent S3D videos [1]. Representing the left and right views separately with full resolution, shown in Figure 1.1(a), is perhaps the most straightforward way. It, however, doubles the data rate of a conventional 2D video. In order to use the existing 2D infrastructure and equipment for 3D compression and transmission, frame-compatible S3D formats are widely used. These formats spatially or temporally multiplex the left and right views into one video stream. Spatial multiplex is

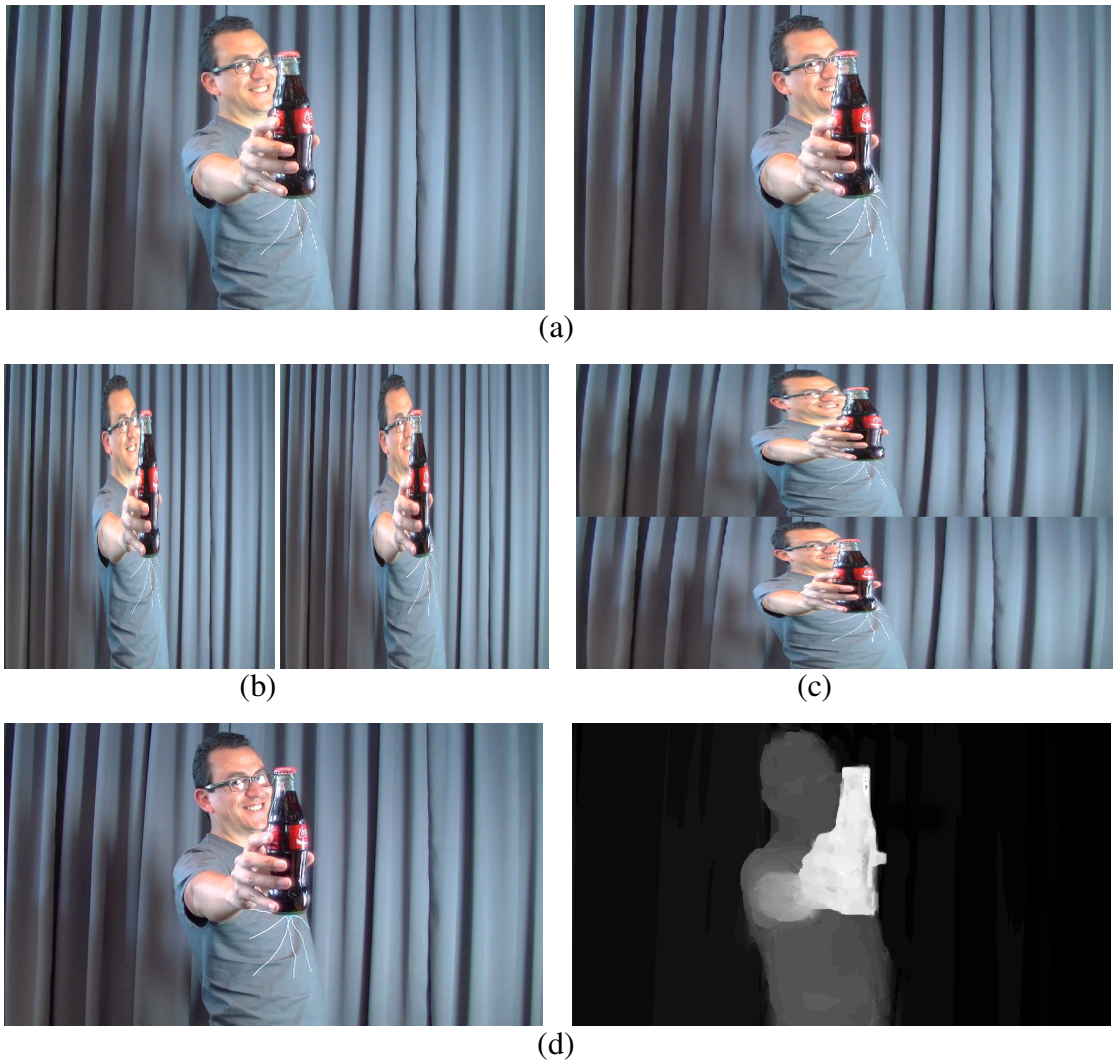


Figure 1.1: Popular stereoscopic 3D formats. (a) Left and right views of the full resolution format, (b) side-by-side frame-compatible format, (c) top-and-bottom frame-compatible format, and (d) 2D-plus-depth format.

typically performed by down-sampling the left and right views horizontally or vertically by two, and putting the downsampled views side-by-side or top-and-bottom, respectively. Examples are shown in Figure 1.1(b) and Figure 1.1(c). The resolution of the integrated video is often that of the 2D video, for example 1920×1080 pixels high-definition (HD) resolution. Temporal multiplex usually interleaves the left and right views as alternating frames or fields. The frame rate of each view may be reduced to keep the data rate equivalent to that of a single view.

Depth-based representations, such as the 2D-plus-depth format shown in Figure 1.1(d), enable the generation of virtual views through depth-based image rendering techniques. The depth map can be computed by matching videos of two or multiple views [2], [3], [4], [5], generated solely based on the 2D video [6], [7], or it can also be directly measured by range cameras [8], [9], [10]. These formats allow adjustment of depth perception in S3D displays based on features such as display sizes and viewing distances, to achieve the best viewing experience.

1.1.2 3D Display Technologies

3D displays show two slightly offset views to the viewer's left and right eyes. There are many techniques for doing this [11]. Many displays pair with special glasses in order to separate multiplexed two views into the left and right views for each eye. These glass-based displays are classified into displays using passive glasses and displays using active glasses.

Passive glasses are again divided into color-multiplexed and polarization-multiplexed approaches. A classic and inexpensive color-multiplexed approach uses the anaglyph glasses. It uses complementary colors, such as red and cyan [12], to multiplex the left and right views. This technique, however, experiences the loss of color information and the increased degree of crosstalk [13], [14]. Another color-multiplexed approach is from Infitec GmbH (Ulm, Germany). It uses two slightly different sets of primary colors (i.e., different wavelengths of red, green, and blue) to project the left and right views. The paired glasses precisely pass one set of wavelengths, ensuring that each eye sees the

correct view [15], [16]. This technique provides full color spectrum and full resolution. It is used in Dolby 3D cinema technology.

A competitive method used in 3D cinema by RealD is the polarization-multiplexed projection system [17]. It multiplexes the two views so that their states of polarization of light are mutually orthogonal. It employs a circular polarization instead of a linear polarization to allow more head tilt before crosstalk becomes noticeable. Compared with the Infitec approach, this method uses low cost passive glasses, but it requires the use of a special screen to effectively control the polarization direction. Another polarization-multiplexed method horizontally interleaves the left and right views on a flat panel display. Then, the interleaved pixel rows are orthogonally polarized by the micropolarizers attached to the display [18]. Low cost passive glasses are used to separate the two views. This, however, results in a reduced vertical resolution.

Active shutter glasses use time-multiplexed approach. The left and right frames are alternatively displayed on a screen at a high frame rate. The liquid-crystal-based glasses temporally alternate between blocking the left eye and the right eye, so that each eye only sees the view intended for it. This requires precise synchronization between the display and the glasses, which makes the glasses battery operated, heavy, and pricy. This technique is currently popular in 3D home theaters.

Contrary to all glasses-based displays, autostereoscopic displays apply parallax barrier or lenticular lens [19], [20], [21] technologies to block or direct light to different directions simultaneously in order to separate views without using any glasses. Such displays can provide two views or multiviews, which provide greater viewing freedom

and are suitable for multiple users. The spatial resolution of each view is reduced according to the number of views provided. Autostereoscopic displays are often found in portable 3D devices such as LG and HTC cell phones, the Nintendo 3DS, and Fujifilm 3D cameras, and multiview displays, which typically provide between five and a few dozen views.

3D displays have many sizes, various optimal viewing distances, and different aspect ratios. Given the same 3D content, special adjustments are needed in order to obtain an optimal viewing experience on a particular 3D display. Adjustment of content aspect ratio will be discussed in detail in Chapter 3 of this thesis.

1.1.3 High Dynamic Range Imaging Technology

Recently, high dynamic range (HDR) displays have gained significant interest in industry. Dynamic range refers to the ratio between the maximum and minimum values of a physical measurement. It is known that the dynamic range of conventional displays is very limited compared to the range of intensities perceivable by the human visual system. Conventional displays usually have a dynamic range of about two orders of magnitude; whereas the human visual system has an overall dynamic range of nearly ten orders of magnitude, and can simultaneously perceive intensities over a range of about five orders of magnitude [46], [67]. This has motivated the development of HDR content capturing and display technology.

Tone mapping (TM) operators [68], [69], [70], [71], [72], [73], [74], [75] convert HDR content to LDR in order to show HDR on a conventional LDR display devices whereas inverse tone mapping (ITM) creates HDR images and videos from LDR content.

Recently developed HDR displays [76] have greatly extended the limited dynamic range of conventional cathode ray tube (CRT), liquid crystal display (LCD), and projector-based displays. Akyüz et al. [77] shows that HDR displays produce pictures with more appealing subjective quality than conventional LDR displays. While new displays tend to offer higher dynamic ranges, HDR capturing technology is still at the early development stages. In order to take advantage of the HDR displays and enable this market, it is necessary to create HDR content from conventional LDR images and videos. This is possible by using efficient inverse tone mapping (ITM) techniques. One challenge that affects the quality of the resulting HDR content in this process is color distortion that is due to saturation in the original LDR content and results in disturbing perceptual artifacts when shown on HDR displays. The problem is more severe for 3D HDR. Hence, designing algorithms that eliminate the LDR to HDR color distortion is of high importance.

1.1.4 Quality Assessment Methods for Visual Media

Image and video quality refers to a characteristic of the visual media that passes through an imaging pipeline. The pipeline, composed of capturing, processing, compression, transmission, and display, may introduce distortion or artifacts to the visual content. Hence, quality assessment is critical to the development and evaluation of the technologies related to image and video content.

Quality assessment methods for visual media are generally classified into objective metrics and subjective evaluations. Objective metrics use mathematical models to approximate the results of subjective quality assessment, providing the advantage of automatic and fast calculation. Quality of experience (QoE), also known as quality of

user experience, is a subjective measurement of a user's perceptual experiences with visual media.

The most traditional and widely used objective metric is the peak signal-to-noise ratio (PSNR). However, PSNR values do not always perfectly correlate with a perceived visual quality due to the complexity of the human visual system (HVS). Recently a number of more complicated and precise metrics were developed, for example VDP [22], SSIM [23], CIELAB ΔE [101], S-CIELAB [102], and FSIM [24]. Furthermore, HDR-VDP [25] and HDR-VDP-2 [26] were developed to evaluate high-dynamic-range (HDR) content.

Most of the objective metrics are designed for conventional 2D content, yet in many situations they are not good indicators of the corresponding subjective quality. The development of quality metrics for new video features, such as 3D and HDR, is still in a preliminary stage. Hence, subjective evaluations are mostly used to measure the quality of 3D content.

Many "subjective video quality measurements" are described in the ITU-T recommendation BT.500 [43]. The idea is to show video sequences to a group of viewers and analyze the viewers' ratings on the quality of each video sequence. Depending on the nature of the testing and the visual content, details of the subjective test procedures may vary greatly.

1.2 Capturing and Displaying Stereoscopic 3D Content

Stereoscopic 3D technology has become one of the main driving forces in the consumer electronics market and the entertainment industry [1]. Hollywood studios are

releasing most of their high-budget movies in 3D and there is a vast selection of 3D TVs available for regular consumers. In addition, other devices capable of displaying stereoscopic content will soon be offered in the market. Many viewers, however, are yet to be convinced of the value of 3D technology. Some of the criticism expressed is directly related to the headaches and nausea, which are more evident when 3D content is viewed on home 3D displays. In order to increase the quality of stereoscopic content for household displays, it is necessary to gain a better understanding of the technical and artistic challenges of this medium. Over the years, stereographers have empirically obtained a few rules of thumb for capturing stereoscopic content [28], [29], [30]. Unfortunately, this pragmatic set of recommendations has not been quantified and there has not been an effort to systematically measure its effectiveness.

1.2.1 Camera Setup

One of the main factors for capturing high-quality stereoscopic content is the proper setup of the two cameras, since it allows content creators to control the 3D effect [30]. There are basically two options for setting up the cameras. The first option is having the cameras converge as shown in Figure 1.2. For this example, the camera axes converge on the little girl. When the 3D video is displayed, the image of the girl will appear on the plane of the screen since, for this object, there is no disparity between the left and right views. In the case of the building, however, there will be a difference along the x (horizontal) axis between the right and left views. This is known as a positive horizontal parallax and it makes the building appear to be behind the screen. Changing the angle of convergence between the cameras can be used to control which objects will pop out from the screen and which objects will remain inside. Unfortunately, this camera configuration

has shown to have side-effects that produce undesirable distortions to the stereoscopic depth [31]. The main distortion caused by this setup, known as the keystone effect [32], creates vertical disparities in the four corners of the screen.

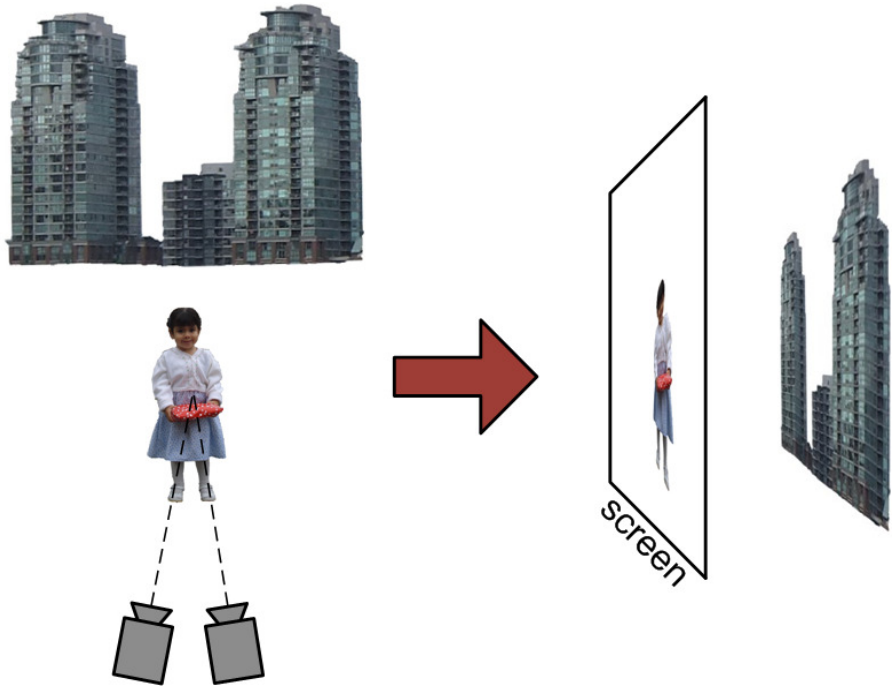


Figure 1.2: Toed-in camera set up for capturing stereoscopic video. The cameras are set up so that their axes converge on a particular object. This object will appear in the plane of the 3D screen.

The second option, which seems to be more popular, consists of setting up the two cameras in parallel (see Figure 1.3). The cameras converge at infinity and the resulting 3D scene appears to be entirely in front of the screen. Each photographed object in this case is known to have a *negative* horizontal parallax. This negative parallax occurs when the left view of an object is located further to the right than the right-view version of the same object. The three different possible types of horizontal parallaxes (negative, zero and positive) are illustrated in Figure 1.4.

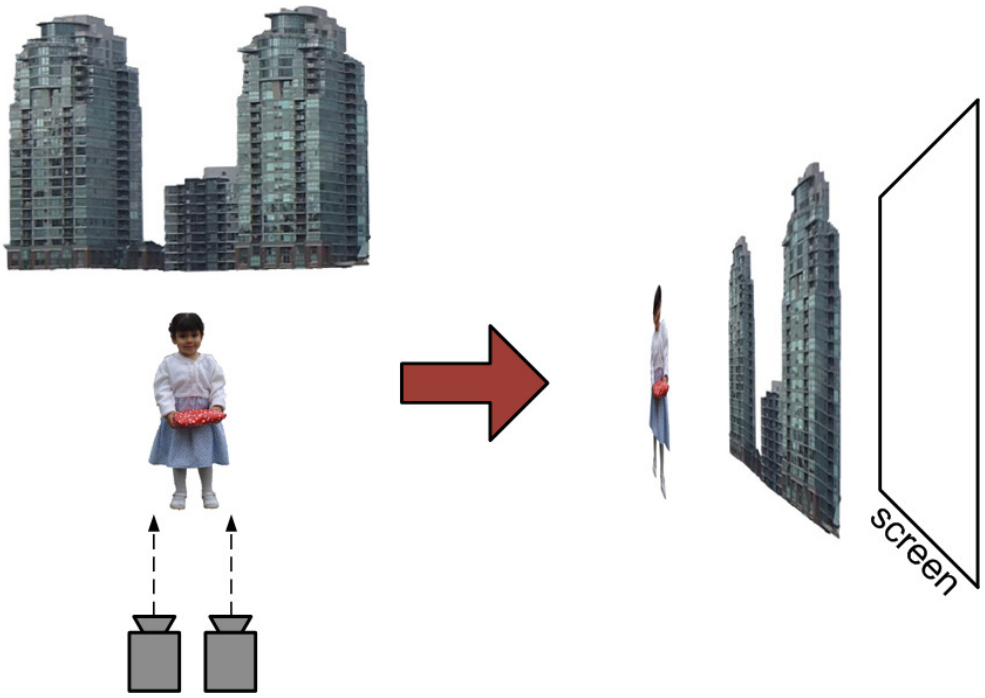


Figure 1.3: Parallel camera set up for capturing stereoscopic video. The cameras are parallel and their axes converge at infinity. All the photographed objects appear to be in front of the screen.

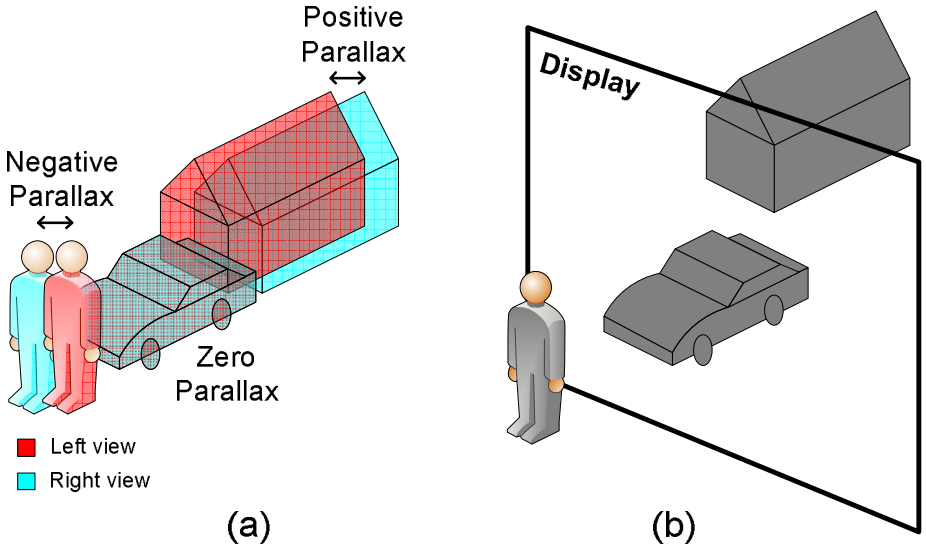


Figure 1.4: (a) Examples of negative, zero, and positive horizontal parallaxes. The red objects are from the left view and the cyan objects are from the right view; (b) when watching 3D content, negative parallax results in objects popping out of the screen, zero parallax positions objects on the screen, and positive parallax results in objects appearing behind the screen.

1.2.2 Horizontal Parallax Adjustment

It is commonly accepted that being exposed for a considerable amount of time to objects that appear to be in front of the screen (i.e., with negative parallax) makes viewers uncomfortable. As described before, when 3D content has been shot with two parallel cameras, all the objects have negative parallax and, therefore, it is a good practice to modify the content in order to reallocate the 3D effect behind the display. In order to do this, it is necessary to modify the depth information that is produced when the 3D content is captured. One solution is proposed in [33], employing an algorithm that modifies horizontal disparity in a nonlinear fashion by warping the input video streams. Unfortunately, for image regions with frequent and strong changes in disparity, this warping scheme can lead to visible distortions. Another way of changing the depth information is by reducing the negative horizontal parallax of 3D videos by shifting the left frames towards the left and the right frames towards the right. Although this action introduces black lines on the vertical edges of the frames, this inconvenience can be sidestepped by cropping the content (to match the aspect ratio) and then scaling it up.

1.3 Reframing Technology Overview

3D-capable devices are currently available to consumers in many aspect ratios. 3D TVs, usually larger than 40”, feature a 16:9 aspect ratio, whereas smaller screens, such as the ones found on tablets and mobile devices, usually have 4:3 and 3:2 aspect ratios, respectively (see Figure 1.5). Stereoscopic 3D media creators tailor their content for a specific aspect ratio (usually 16:9). Unfortunately, playing this content on 3D displays

with aspect ratios that are different to the intended one may degrade the quality of the viewing experience.

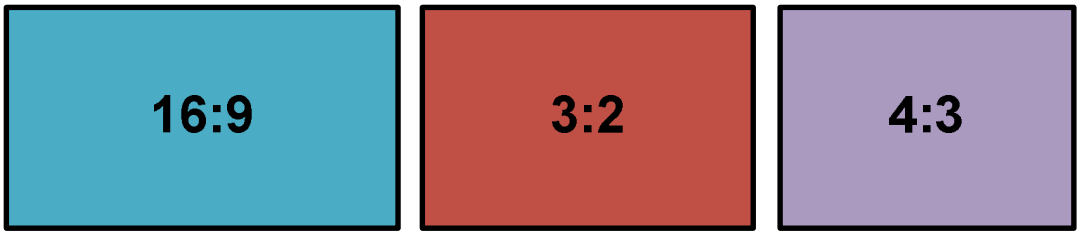


Figure 1.5: Different aspect ratios.

1.3.1 Existing Reframing Solutions

Several solutions have been proposed in order to compensate for this variation in aspect ratios (see Figure 1.6). The straightforward option is to add black bars to the screen, which can be horizontal (also known as letterboxing) or vertical (also known as pillarboxing), depending on the original and new aspect ratios [45]. The main problem with this option, as illustrated in Figure 1.6(b), is that a significant part of the screen will remain unused. This is particularly problematic for small devices, since important parts of the frames become too small to see. A second option, exemplified in Figure 1.6(c), consists of squeezing the content so that it fits within the new aspect ratio (anamorphic video). Another alternative is cropping the borders of the video frames so that the modified frames have the proper aspect ratio (see Figure 1.6(d)). This technique, known as centered cropping, eliminates visual information without taking into account that these regions might actually be of interest to the viewers.

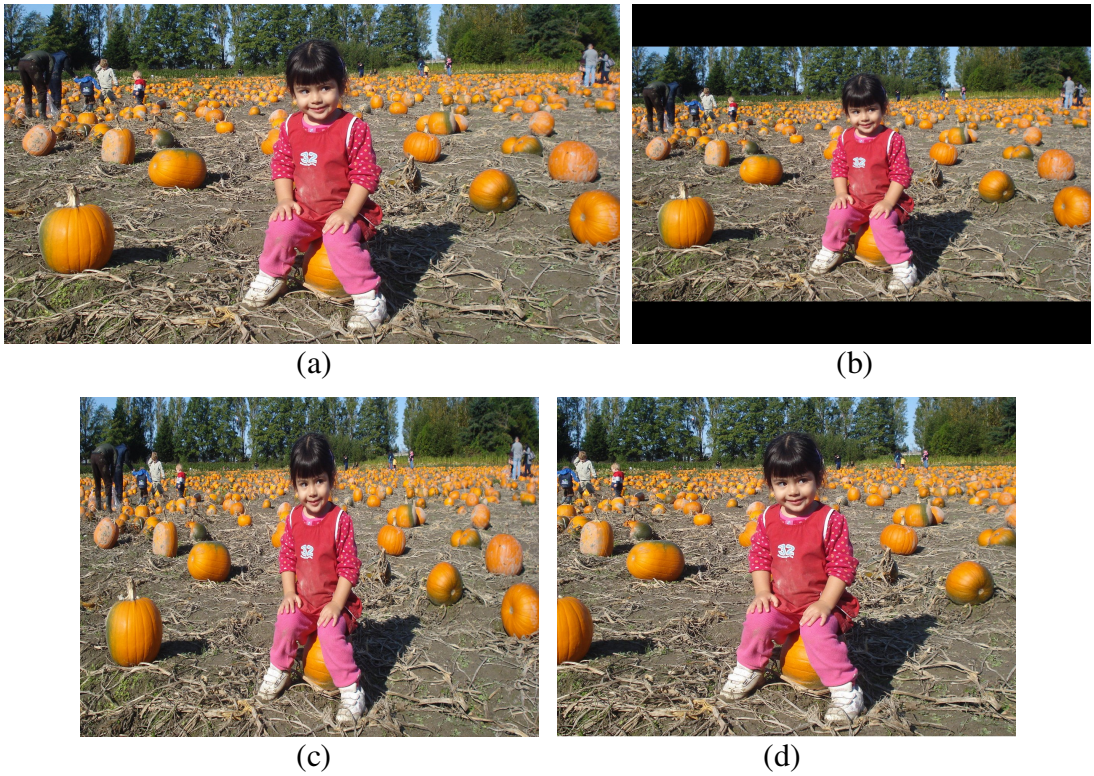


Figure 1.6. Different methods for displaying a video frame with a 16:9 aspect ratio on a screen with a 4:3 aspect ratio: (a) a video frame with a 16:9 aspect ratio, (b) letterboxing method, (c) anamorphic method, and (d) centered cropping method.

An alternative solution is to involve humans throughout the reframing process. Human observers can detect the important visual points on the screen and control the location of the bounding box (i.e., the region of the frame that will prevail after the reframing process). This process, known as pan and scan, ensures that the modified content will be meaningful to the viewers, but it is evidently expensive, time-consuming, and not suitable for real-time applications. A better solution is to have an automatic process that identifies the main visual information and keeps it inside the bounding box.

1.3.2 Visual Attention Model and Automatic Reframing Techniques

Several methods have been proposed for automatic content reframing. A vast majority of these schemes, however, deal exclusively with 2D still images [46], [47],

[48]. For the case of automatic 2D video reframing, [49] and [50] propose schemes that aim to preserve visually important regions as well as temporal stability. The Visual Attention Model (VAM) used in [50] is taken from [51], [52]. Two Kalman filters are used to ensure good temporal consistency. The objective of the filters is to smooth the change in the values of the bounding box center coordinates on every frame.

Color and depth are employed to create a visual attention model for 3D images in [53]. Results and conclusions, however, were drawn using data from merely five stereo images.

An early proposal for a visual attention model for 3D video is found in [54]. The proposed scheme uses cues such as stereo disparity, image flow, and motion. Relative depth was employed as a target selection criterion. This scheme is able to detect the moving object that is closest to the cameras. Although this solution might be useful for some videos, it will not provide acceptable results for complex scenes like the ones usually found in commercial videos made by the entertainment industry.

Another VAM for 3D video is presented in [55]. The model uses features such as depth information, luminance, color, and motion. This scheme, however, was developed and tested on multiview videos, and the disparity is computed based on a graph cuts algorithm for multiview reconstruction. It cannot be directly applied to stereoscopic video content. This VAM also uses some very computational expensive steps, which is unsuitable for near real-time applications.

1.4 Color Distortion Due to Saturation in Low Dynamic Range Content

Legacy images and videos store only a small dynamic range of information due to the limitations of the capturing and display devices. The very bright or dark parts of a scene are clipped to the upper or lower displayable limits, and as a result information is lost. Special attention has been paid to the restoration of the clipped pixels. Over the last few years, several methods [78], [79], [80], [81], [82] have been developed to enhance the luma of the clipped pixels, so that the enhanced clipped regions have higher dynamic range and look more realistic on HDR displays.

1.4.1 Existing Algorithms for Saturated-Pixel Enhancement

Meylan et al. [79] apply a simple piecewise linear tone scale function, composed of two slopes, one applied to the diffuse areas and one applied to the specular reflected areas, in order to particularly enhance the specular highlights. In [80], the Median Cut algorithm and Gaussian filter are applied to estimate the light source density map. Then a luma expansion is applied to enhance the clipped regions. In [81], a smooth brightness enhancement function is obtained by blurring a binary mask with a large kernel of approximately Gaussian shape. A semi-automatic classifier was developed in [82] to classify the clipped regions as lights, reflections, or diffuse surfaces. Each class of objects is enhanced with respect to its relative brightness. All of the above schemes enhance only the luma, while the chroma enhancement is not considered.

For a clipped pixel, often not all three red (R), green (G), and blue (B) channels are clipped, nor does the same amount of clipping occur in each channel. If clipping changes

the *R*, *G*, *B* color ratios of a pixel, then the result is color distortion. In fact, color distortion occurs very often. Although people are accustomed to the clipping effect in highlights, where the distorted color is desaturated and close to white, the distorted colors near the midtone produce a very noticeable and disturbing effect. Figure 1.7 gives an example of color distortion caused by the clipping effect. Figure 1.7(a) is a correct-exposure image, and Figure 1.7(c) is the corresponding over-exposed image, where the yellow pixels reveal obvious color distortions due to the clipping. We enhance the luma of the clipped image using the best possible values (i.e., the ground truth luma of the correct-exposure image), while keeping the chroma distortion unchanged. The luma-enhanced image is shown in Figure 1.7(e). Figure 1.7(b), (d), and (f) depict the plots of RGB intensities versus the pixel-position along the same horizontal line shown in Figure 1.7(a), (c), and (e), respectively. From Figure 1.7(c) and (d), we observe that the red channel is saturated for most pixels on the shoe (non-shoelace part); and for this reason this part appears yellow in Figure 1.7(c) rather than being orange as in the correct-exposure picture (in Figure 1.7(a)). As shown in Figure 1.7(e), enhancing only the luma does not solve the color distortion problem. In fact the color distortion in this image appears at least as disturbing as in the clipped image (Figure 1.7(c)). Ideally, instead of changing all the color channels as it is done by luma enhancement, we want to enhance only the clipped color channel(s). Figure 1.7(f) shows that enhancing luma makes the unclipped channels (i.e., *G* and *B*) less accurate. Therefore, we need an algorithm that corrects color and at the same time enhances luma of a clipped image.

A few methods were developed to enhance the color saturation and fill in clipped regions. One category of the enhancement methods is to remove the specularity from the

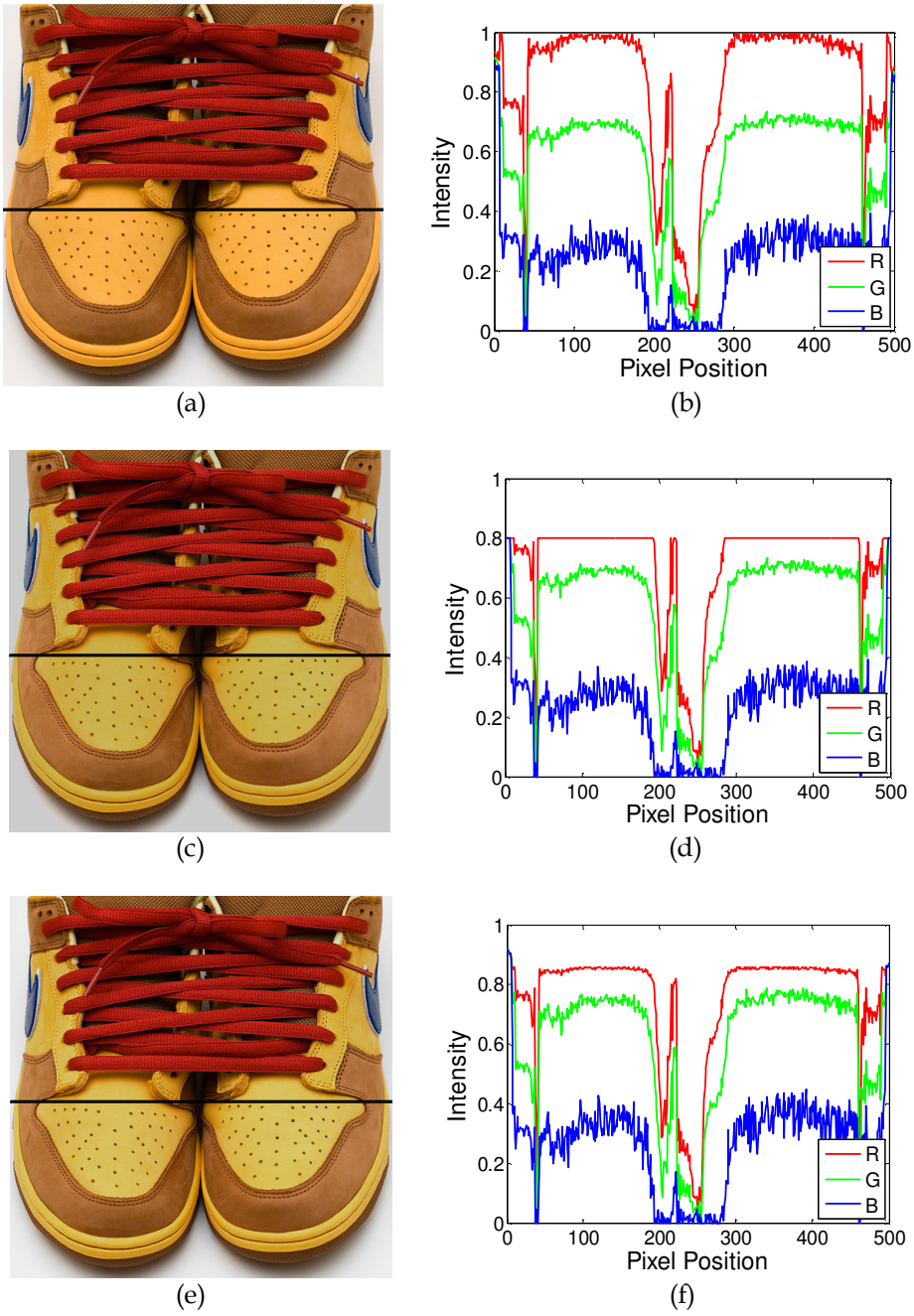


Figure 1.7: Example of color distortion due to clipping. (a) A correct-exposure image, (c) the corresponding over-exposed image, and (e) the enhanced image with corrected luma for clipped pixels. (b), (d), and (f) are the plots of the RGB intensities versus the pixel index along the highlighted horizontal lines of images (a), (c), and (e), respectively.

highlighted area, and reconstruct the highlighted object assuming only diffuse reflection exists [83], [84], [85]. These methods are particularly useful as a pre-processing before tasks such as stereo matching, segmentation, and object recognition. The highlight-

removed images, however, do not reflect the real scene under the original lighting. Hence, these approaches cannot restore the lost information due to clipping. Wang et al. [86] proposed an effective HDR image hallucination approach for adding HDR details to the over-exposed and under-exposed regions of an LDR image. The method assumes that high quality patches exist in the image with similar textures as the regions that are over or under-exposed. It corrects both the luma and the chroma, while it fills in detailed textures for the over-exposed and under-exposed regions. This approach, however, is semi-automatic and it needs the user's input to identify textures that can be applied to fill in the under-exposed or over-exposed areas. This manual intervention is often undesirable.

An automatic method proposed by Zhang and Brainard (ZB) [87], which we shall henceforth refer to as the ZB algorithm, uses a statistical approach to fix saturation for over-exposed pixels in color images. This work exploits the correlation between the responses of RGB color channels at each pixel, and estimates the clipped color channel(s) using the Bayesian algorithm based on the corresponding unsaturated color channel(s) and the prior distribution parameters. The algorithm has low computational cost, and is effective when the statistical properties of the saturated regions are consistent with that of the unsaturated region in the image.

In what follows, we briefly describe the ZB algorithm, since it is the state of the art desaturation algorithm as well as the starting point of the algorithm proposed in Section 4.1. The ZB algorithm uses Bayesian framework for estimating the true values of the saturated pixels. The joint distribution of the RGB color channels was used as the prior

information. A multivariate normal distribution model is used to model the relation among the R, G, and B channels as follows:

$$\begin{pmatrix} X_s \\ X_k + e_k \end{pmatrix} = \begin{pmatrix} X_s \\ Y_k \end{pmatrix} \sim N\left(\begin{pmatrix} \mu_s \\ \mu_k \end{pmatrix}, \begin{bmatrix} V_s & V_{sk} \\ V_{ks} & V_k + V_{e_k} \end{bmatrix} \right) \tag{1.1}$$

where X_s is an $n_s \times 1$ vector of the true values of the saturated color channel(s), X_k, Y_k, and e_k are $n_k \times 1$ vectors of the true values, measured values, and measurement errors of the unsaturated color channel(s), and $n_s + n_k = 3$. The mean values of X_s and X_k are represented by μ_s and μ_k, respectively. The variances of X_s, X_k, and e_k are V_s, V_k, and V_{e_k}, respectively. The V_{sk} denotes the covariance between X_s and Y_k, and we have $V_{sk} = V_{ks}^{\ T}$.

Given the measured pixel values $Y_k = k$ of the unsaturated color channel(s), the conditional distribution $P(X_s \mid Y_k = k)$ of the saturated channel(s) is normal with a mean μ_{x_s} and a variance V_{x_s} [96], shown in (1.2) and (1.3):

$$\mu_{x_s} = \mu_s + V_{sk}(V_k + V_{e_k})^{-1}(k - \mu_k), \tag{1.2}$$

$$V_{x_s} = V_s - V_{sk}(V_k + V_{e_k})^{-1} V_{sk}^{\ T}. \tag{1.3}$$

Then, the saturated channel(s) X_s is estimated by computing the expected value of the posterior distribution, as follows:

$$E[X_s \mid Y_k = k, Y_s \geq s], \tag{1.4}$$

which can be calculated given (1.2) and (1.3).

The conditional variance V_{x_s} of the saturated color channel(s) X_s is smaller than the unconditional variance V_s. This variance reduction and hence more accurate value estimation is due to the additional available information provided by the strong channel correlation V_{sk}.

Based on (1.3), a large covariance V_{sk} results in a small V_{x_s}, which means small estimation uncertainty. The correlation between color channels is the key in the ZB algorithm to estimate the saturated color channel(s) based on the unsaturated color channel(s). The correlation between color channels, however, varies within an image, especially between regions of different chroma. The ZB algorithm, on the other hand, uses the same statistical model to correct all saturated areas in an entire image. Adapting the model to reflect local correlation may improve the desaturation performance.

1.5 Thesis Outline

In this thesis, we present novel methods for improving the quality of viewing experience for stereoscopic 3D content. This includes providing 3D capturing and display guidelines, automatic 3D video reframing, and color correction for saturated 3D content.

In Chapter 2, we provide guidelines for 3D image and video capturing through a series of systematic subjective evaluations. We present a comprehensive 3D image and video database with the content captured at various distances from the camera lenses and under different lighting conditions. We conducted subjective tests to assess the perceived 3D quality of these videos and images which were shown on 3D displays of different

sizes. In addition, we adjusted the horizontal parallax of the content to verify and quantify via subjective tests whether and how much this change could increase the viewer's quality of experience. Finally, we provide guidelines of acquisition distances between the cameras and the real scene.

In Chapter 3, an automatic content-aware reframing solution for stereoscopic 3D video is presented. Since 3D displays have various aspect ratios (e.g., 16:9, 4:3, and 3:2), watching 3D videos with the wrong aspect ratio decreases the quality of the viewing experience. We have developed a smart reframing solution that uses a visual attention model for stereoscopic 3D video to identify the prominent visual regions of every stereoscopic frame. Our method uses several saliency indicators such as depth, edges, motion, brightness, and color. Additionally, our method provides a dynamic bounding box that avoids annoying reframing issues, such as jittering and window violation. Experimental results over a great variety of videos show that our proposed reframing algorithm is both effective and robust.

Finally, in Chapter 4 we present two methods for correcting the color-saturation problem caused by the limited dynamic range of conventional low dynamic range content. The color-corrected content can be shown on a high dynamic range display, through inverse tone mapping. In Section 4.1, we present a Bayesian-based color desaturation algorithm. Unlike the previously proposed state-of-the-art Bayesian algorithm, which uses all unsaturated pixels in an image to estimate the prior distribution, our method uses local statistics for correcting each disconnected saturated region. Our method utilizes the inter-channel correlation as well as the strong spatial correlation of

images. Experimental results show that our method results in a significant improvement over the state-of-the-art color-desaturation algorithm.

Both the state-of-the-art algorithm and the algorithm we proposed in Section 4.1 use the correlations between R, G, and B color channels, which may not be the most suitable way for exploiting the relationships among color pixels. The pixels could be more strongly correlated in the spatial domain and in some other color spaces. In Section 4.2, we propose another effective color-desaturation algorithm, which exploits the strong correlation in chroma between saturated pixels and their surrounding unsaturated pixels. The algorithm automatically restores both the luma and chroma of the clipped pixels. Extensions to videos and 3D content are also proposed for wider applications. Experimental results show that this algorithm outperforms both the state-of-the-art algorithm and the algorithm proposed in Section 4.1 in both objective and subjective quality evaluations, yielding 2D and 3D content with higher dynamic range and vivid color.

2 Guidelines for an Improved Quality of Experience in 3D TV and 3D Mobile Displays

Stereoscopic 3D movies have become widely popular all over the world. In addition, 3D TVs and mobile devices have already been introduced to the consumer market. However, while some manufacturers are introducing 3D cameras and movie studios are using proprietary solutions, there are no guidelines for consistently capturing high quality stereoscopic content. As a result, problems such as headache and visual fatigue are preventing 3D technology from being widely adopted and compete with its conventional 2D counterpart. Having guidelines for capturing and displaying 3D images and videos will result in improved 3D quality of experience and hence boost broad adoption of 3D technology by the consumer market. In this study, we tested the effect that different distances (measured from the 3D camera setup to the photographed objects) and lighting conditions have on the quality of the stereoscopically captured images and videos when viewed on home 3D TVs and 3D mobile devices.

Developing a reliable objective quality metric for 3D content has proven to be very challenging [34], [35]. Therefore, researchers have mainly relied on subjective evaluations such as [36], [37], [38], [39], [40] to identify the key factors for producing high-quality stereoscopic content,

In order to successfully assess user experience, the opinion scores must be taken from an adequate sample of typical users carrying out representative tasks in a realistic context of use [41]. Because of this, more meaningful results will be obtained if the media employed for these tests resembles content that is actually being shown on 3D TVs (i.e.,

featuring people and objects in ordinary surroundings instead of an artificial lab setting). Testing both images and video sequences is also desirable since spectators might perceive quality differently for different types of content.

We have created our own stereoscopic image and video database that is comprised of scenes depicting people and landscapes with various distances between the cameras and the subjects. Our subjective assessment exercise is comprised by three stages. During the first stage, several viewers of different ages watched and rated the stereoscopic images that we captured using various distances between the cameras and the subjects. For the second stage, the content consisted on stereoscopic video sequences that were shot using the same combination of distances as in the previous test. Viewers were asked again to rate the 3D quality of the content. Finally, we performed subjective evaluations to verify and quantify the influence in 3D quality of experience caused by the adjustment of the horizontal parallax.

The rest of the chapter is organized as follows. Section 2.1 describes the 3D content acquisition and alignment processes. The subjective evaluation environment and parameters are specified in Section 2.2. In Section 2.3, we present the statistical analysis of the subjective test scores and discuss the findings of the tests. We conclude the chapter in Section 2.4.

2.1 Acquisition and Alignment

2.1.1 Equipment

In order to capture stereoscopic video and images we employed two identical HD cameras with the same firmware and settings. These cameras were aligned in parallel and

attached to a bar that was specifically made for them. Subsequently, the bar was secured to a tripod as shown in Figure 2.1. Since zoom lenses may differ [37], only an extreme end of the zoom range was used. A single remote control was employed to start both cameras simultaneously and obtain the best possible synchronization.

Figure 2.1: Stereo camera setup consisting of two identical HD camcorders.

2.1.2 Image and Video Capturing

The stereoscopic video and image capturing process is illustrated in Figure 2.2. Both cameras capture slightly different images of the same event. Each event on our database consists mainly of a person or object standing in front of the camera with a wall or a building as background. For all videos, the object of interest is the one that is closest to the cameras. The camera is always kept still while the people and some objects move moderately. There are four important distances that need to be considered for every stereoscopic image or video pair. They are described in Table 2.1.

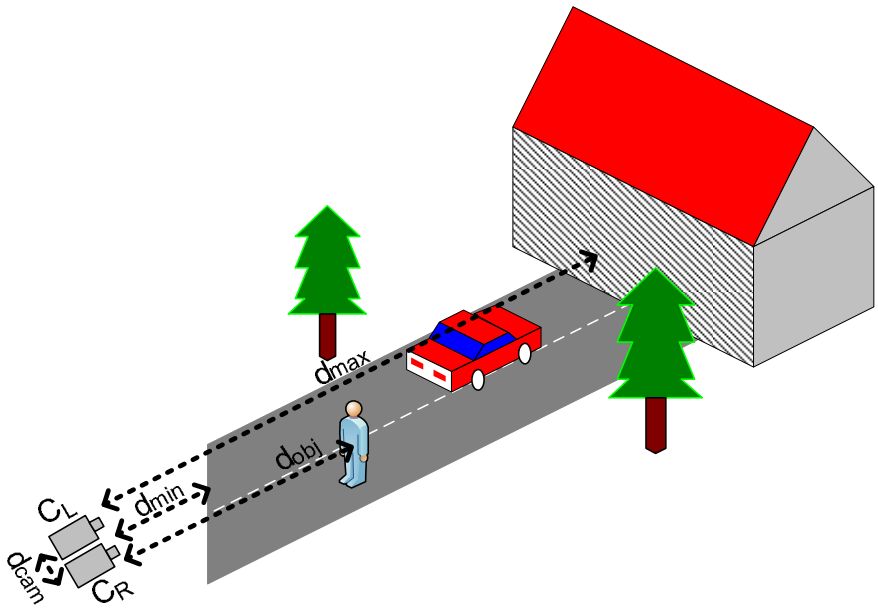


Figure 2.2: Capturing a live-action event with two parallel cameras C_L and C_R; d_{cam} is the distance between the cameras, d_{min} is the distance from the cameras to the closest point, d_{obj} is the distance from the cameras to the main object (usually a person), and d_{max} is the distance from the cameras to the background.

Table 2.1: Distances considered when capturing stereoscopic images and videos for our database

Distance	Description	Values
d_{cam}	Distance between the two cameras	77 mm
d_{min}	The distance between the cameras and the closest point captured in the stereoscopic image or video pair	0.5 m, 1 m, 2 m, 3 m
d_{obj}	The distance between the cameras and the main object (usually a person). In most cases, $d_{obj} = d_{min}$	0.5 m, 1 m, 2 m, 3 m
d_{max}	The distance between the cameras and the furthest background. If the sky is visible, then d_{max} is considered to be infinity	5 m, 10 m, 50 m, infinity

2.1.3 Temporal Synchronization

Three out of thirty videos were unsynchronized by a few frames and were manually synchronized before further processing.

2.1.4 3D Content Alignment

Even though the cameras are carefully lined up, a small amount of vertical disparity between the left and the right views of a stereoscopic image/video is unavoidable. Therefore, it is usually necessary to vertically align the left and right views. This alignment process is performed for every stereoscopic image and video pair in our database.

In addition, for the third stage of our quality assessment process, we also eliminate the negative horizontal parallax so that the photographed objects do not appear to pop out of the screen. This alignment process involves horizontal frame shifting. By eliminating all negative disparities we avoid stereoscopic window violations [29] (i.e., when objects that pop out of the screen are only partially shown thus providing the brain with two conflicting depth cues). However, eliminating negative disparities through shifting may result in large positive disparity regions in the background of a scene, which cause eye divergence. Since the background is usually not the point of interest, especially in the case of video, such divergence does not have a strong negative impact to the quality of the content. This is later verified by the results of our subjective tests in Section 2.3.5.

Both vertical and horizontal alignments involve frame shifting and are performed automatically using features that are common to both left and right videos. The objective is to be able to implement this method on 3D TV displays and achieve this "correction" in real-time. Our method uses the Scale Invariant Feature Transform (SIFT) algorithm to identify matching features on both views [42]. Having the coordinates to these common

features allows us to compute the parallax between the left and right views of the photographed objects. The following steps describe in detail this algorithm:

1. The first frame of both the left and right videos is downsampled by a factor of 2 (horizontally and vertically) to reduce the number of computations in the next steps and allow real-time implementation.

2. The features of the downsampled left and right frames are obtained using SIFT.

3. The features of the left frame are matched to the features of the right frame. The top ten percent of all matching features, whose vertical disparities are considerably different from the median disparity value of all matching features, are detected as outliers. These outlier features are removed to ensure the stability of the algorithm. The Cartesian coordinates of rest of the matching features are stored.

4. Δy, the amount of pixels that each original frame will be shifted vertically, is found by computing the median of all the y coordinates of matching points between the two frames and then multiplying by 2 (to compensate for the downsampling).

5. Δx, the amount of pixels that each original frame will be shifted horizontally, is computed by finding the largest negative value of all the x coordinates of matching points between the two frames and then multiplying by 2 (to compensate for the downsampling). The negative number with the largest absolute value of the x coordinates represents the photographed point in space that is closest to the cameras (d_{min}).

6. Finally, the shifted frames are cropped and then enlarged using bicubic

interpolation so that they maintain the same size they had before the shifting process (1080 pixels × 1920 pixels).

An example of a 3D video frame before and after the shifting algorithm is shown in Figure 2.3(a) and Figure 2.3(b), respectively. For illustration purposes, the stereoscopic frames are displayed in anaglyph (red and cyan) mode. Notice how the superimposed left and right frames are vertically misaligned in Figure 2.3(a). This problem has been solved in Figure 2.3(b). In addition, the two frames have been horizontally shifted to produce a zero parallax for the closest object (in this case, the photographed individual); the relative positions of the objects behind the subject indicate positive parallax.

(a) (b)

Figure 2.3: A stereoscopic video frame from an indoor sequence with d_{min} = 3 m, d_{max} = 5 m and presented in anaglyph mode for illustration purposes: (a) without any vertical and horizontal shifting, (b) after both vertical and horizontal shifting plus cropping and interpolation to preserve the 16:9 aspect ratio.

2.2 Evaluation Environment

2.2.1 Displays

Our subjective tests were conducted on four different sizes of stereoscopic displays, namely, a 2.8” 3D camera display, a 22” 3D LCD display, a 55” 3D LED TV, and a 65” 3D Plasma TV. The 2.8” display is an autostereoscopic display that can be viewed

without glasses, and the other three displays are paired with different 3D active shutter glasses. The detailed specifications of the four displays are listed in Table 2.2.

Table 2.2: Properties of the 3D displays used in our test

Size	Resolution	Refresh Rate	Glasses
2.8”	Approx. 230,000 dots	-- --	No glasses needed
22”	1680 x 1050	120Hz	3D active shutter glasses
55”	1920 x 1080	240Hz	3D active shutter glasses
65”	1920 x 1080	600Hz	3D active shutter glasses

2.2.2 Database

The images have a resolution of 3840 pixels × 2160 pixels, and the videos have a resolution of 1920 pixels × 1080 pixels and frame rate of 30 frames per second. The database includes thirty images and thirty video sequences with various combinations of d_{min}, d_{obj}, and d_{max}, where d_{min} is in {0.5, 1, 2, 3} meters, d_{obj} is in {0.5, 1, 2, 3} meters, and d_{max} is in {5, 10, 50, infinity} meters. We also prepared a different set of ten images and four videos as training sequences for our test. All images and videos were shot in a natural environment rather than a lab setup. The main objects in the scenes are often people, chairs, toys, and buildings (see a sample in Figure 2.4).

2.2.3 Observers

Nineteen observers participated in the first stage of our test, including six females and thirteen males. Their ages ranged from 23 to 59, with an average age of 33. In the second and third stages of our test, another twenty subjects participated, including seven females and thirteen males. The average age was 32. All observers are non-expert in viewing 3D images and videos, and they were screened for visual acuity using the Snellen chart and color vision using the Ishihara test.

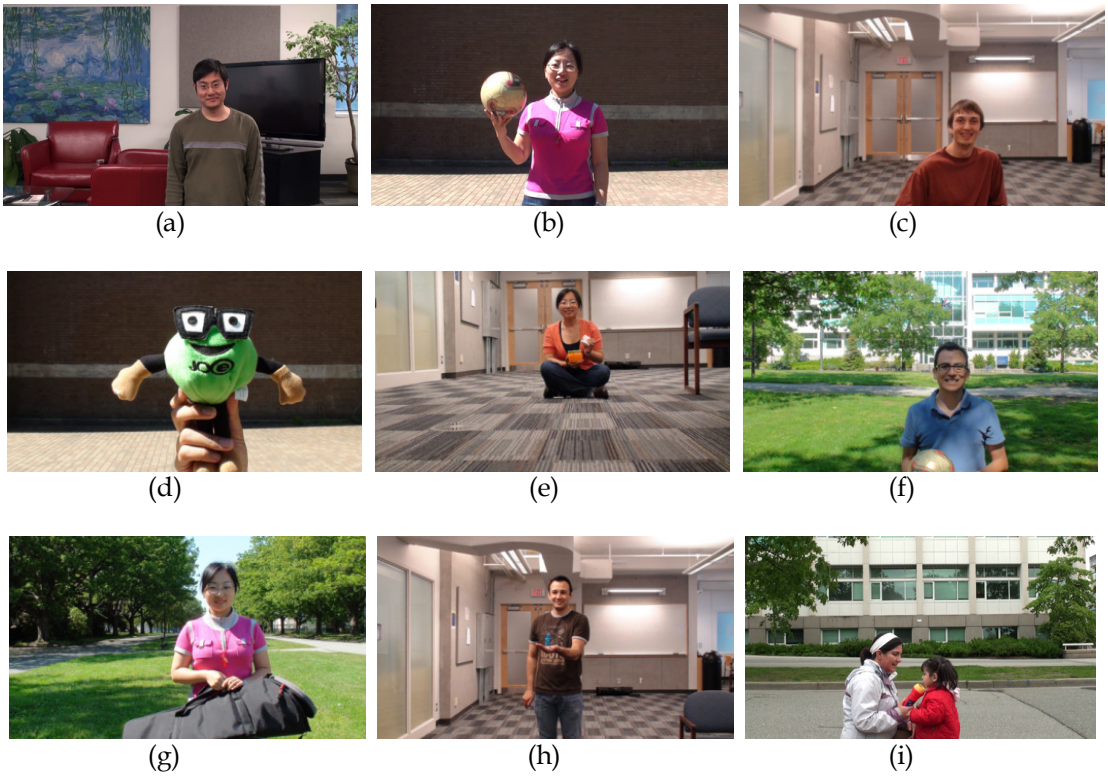


Figure 2.4: The left view of some images and video frames from our 3D database.

2.2.4 Testing Procedure

We set up the viewing conditions for the subjective assessment according to Section 2.1 of the ITU-R BT.500-11 [43]. A single stimulus method has been adopted for the subjective quality evaluation. Before the subjective evaluation on each display, we ran a training session to show to the subjects the quality range of our stereoscopic images and videos, without imposing the quality of the content.

In the first stage of our test, thirty test images were used and were shown in a random order on three displays, i.e., the 2.8" 3D camera display, the 22" 3D display, and the 55" 3D TV. During the test, each stereoscopic image was shown for five seconds followed by a five-second interval of a 2D mid-grey image with the image index as a grading and

relaxation period. One to three observers participated in each viewing session. In the second and third stages of our test, sixty ten-second test video sequences were shown on the 65" Plasma TV, with a four-second break of a 2D mid-grey image as a grading and relaxation period. The sixty videos are two versions of the thirty videos in our database. One version is processed for vertical alignment, and the other is processed by applying horizontal and vertical shifts so that the closest object is at the depth of the screen. The order of the sixty videos is randomized so that videos with similar capturing parameters are inconsecutive and the two versions of the same video are kept far from each other. Two observers conducted the test in each testing session. In all of our tests, the viewers were seated in line with the center of the display, and at the distances that were recommended by the manufactures of the displays.

2.3 Analysis and Results

2.3.1 Detection of the Outliers

Before analyzing the scores provided by the observers, we first detect the outliers according to the subjective scores they gave. The screening process is based on the guidelines provided in Section 2.3.1 of annex 2 of ITU-R BT.500-11 recommendation [43]. For each image or video, we first determine the normality of its score distribution by computing the kurtosis coefficient, which is defined as the fourth moment about the mean divided by the square of the second moment about the mean minus 3. In other words, the *j*th kurtosis coefficient is: $kurtosis_j = \frac{m_4}{(m_2)^2} - 3 = \frac{\frac{1}{M}\sum_{i=1}^{M}(x_{ij} - \bar{x}_j)^4}{\left(\frac{1}{M}\sum_{i=1}^{M}(x_{ij} - \bar{x}_j)^2\right)^2} - 3$, where x_{ij} is the score of the *j*th image or video from the *i*th observer and $\bar{x}_j$ is the average score of the *j*th

image or video over all *M* observers. The score distribution is considered normal if -1 ≤ kurtosis$_j$ ≤ 1−1 ≤ kurtosis$_j$ ≤ 1, and non-normal otherwise. To check if the *i*th observer is an outlier, we initialize two counters P_i and Q_i to zeros. The counter values are then updated based on the score x_{ij} (for all *i*), as follows: $\begin{array}{ll} \text{if } x_{ij} \geq \bar{x}_j + c_j\sigma_j & \text{then } P_i = P_i + 1, \\ \text{if } x_{ij} \leq \bar{x}_j - c_j\sigma_j & \text{then } Q_i = Q_i + 1, \end{array}$ where $c_j = 2$, if the score distribution of the *j*th image or video is normal; and $c_j = \sqrt{20}$, otherwise. σ_j is the standard deviation of the scores of the *j*th image or video. Finally, if $\frac{P_i + Q_i}{N} > 0.05$ and $\left|\frac{P_i - Q_i}{P_i + Q_i}\right| < 0.3$, where *N* is the number of test images or videos, the observer *i* is considered as an outlier.

One out of nineteen observers was detected as the outlier in the first stage of our test, and another one out of twenty subjects was detected as the outlier in the second and third stages. All the scores of these outliers were eliminated from the subsequent calculation. Therefore, the data analysis in the three stages is based on the scores provided by eighteen, nineteen, and nineteen valid observers, respectively.

2.3.2 Score Computation

We take the average score across all valid observers for each image or video as the mean opinion score (MOS). To assess the credibility of the mean opinion score, we use confidence intervals to indicate the reliability of an estimate. The Student's t-tests [44] are used to compute confidence intervals with the significance level being 95%.

In the following three sections, we will present the results of our three experimental stages. We will analyze the influence of capturing parameters (i.e., lighting conditions,

d_{min}, d_{max}, and d_{obj} not being the foreground object) to 3D image quality, 3D video quality, and 3D video quality after horizontal parallax adjustment.

2.3.3 Stage One: Influence of Capturing Parameters to 3D Image Quality on Three Sizes of Displays

We first analyze the quality scores from the first stage of our experiment and reveal how the quality of 3D images is affected by the lighting condition, d_{min}, d_{max}, and d_{obj} not being the foreground object.

2.3.3.1 Influence of lighting condition to image quality

Our 3D image database includes eight sets of images. Each set was captured with the same distance parameters (i.e., d_{min}, d_{obj}, and d_{max}) but under two different lighting conditions (outdoor on a sunny day and indoor). For every image, our cameras provided the best exposure parameters. However, indoor images tend to have a more uniform light distribution whereas outdoor images have some bright regions that contrast with some dark ones. We grouped these images by their lighting conditions and the mean opinion scores of the indoor images and outdoor images are compared in Figure 2.5. Indoor lighting results in slightly higher 3D quality than the outdoor lighting for all three sizes of displays. The quality difference between these lighting conditions, however, is insignificant. Therefore, in the following subsections, the mean opinion score of each capturing-parameter set is the average score over the indoor and outdoor scenes.

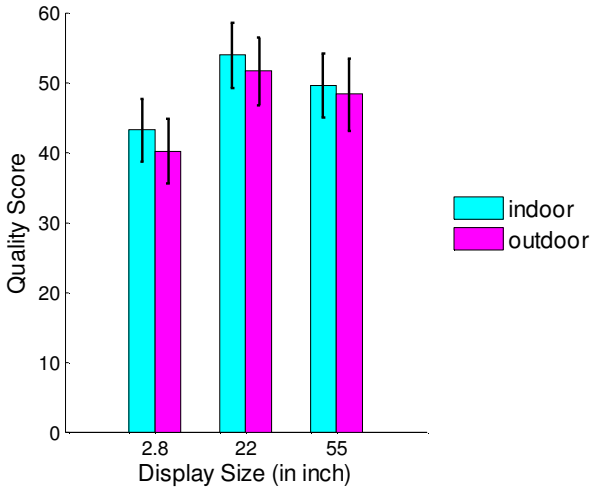


Figure 2.5: The mean opinion scores and their confidence intervals versus different sizes of 3D displays under different lighting conditions.

2.3.3.2 Influence of d_{min} to image quality

We compare the subjective quality between images taken at different d_{min} when d_{min} is the same as d_{obj}. Figure 2.6 shows the mean opinion scores and confidence intervals versus d_{min} at different d_{max} distances. The figure indicates that for the same d_{max}, the image quality increases with d_{min} and levels off when d_{min} is beyond two meters. The confidence intervals when d_{min} is 0.5 meters are smaller than those when d_{min} is large. In other words, the observers consistently provided low scores when the closest object was very close to the cameras. Figure 2.6 shows the results based on scores from the three displays. The quality trend affected by d_{min} is the same for all three sizes of displays.

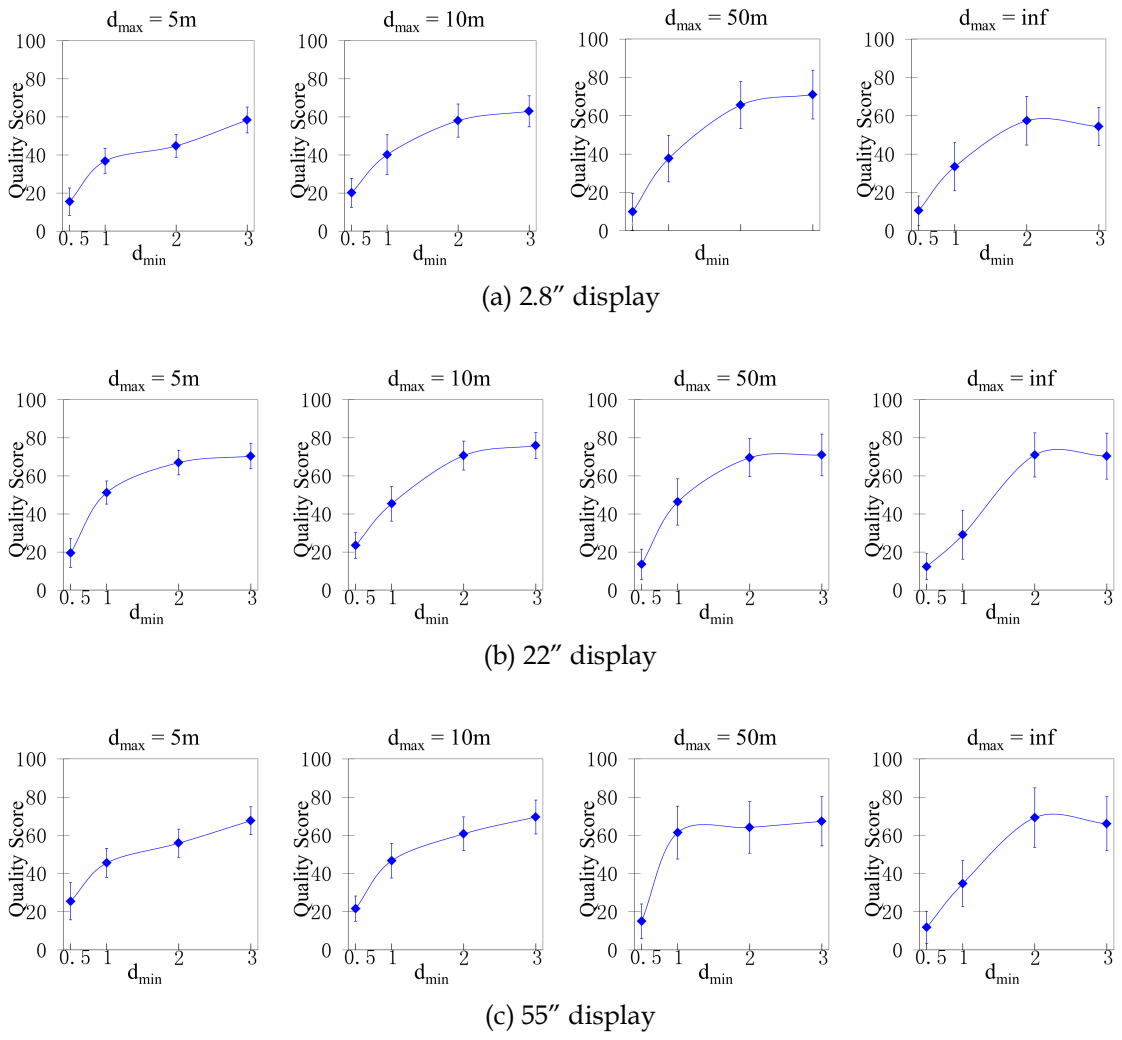


Figure 2.6: The mean opinion scores and their confidence intervals on different sizes of displays at various d_{min} (0.5m, 1m, 2m, and 3m). Parts (a), (b), and (c) show the results associated with the 2.8", 22", and 55" displays, respectively. The four subplots correspond to the cases when d_{max} are 5m, 10m, 50m, and infinity.

2.3.3.3 Influence of d_{max} to image quality

We compare the quality scores between images with different d_{max} while keeping the same d_{min}. No clear trend is observed from Figure 2.7. Thus, we conclude that d_{max} does not strongly affect the quality of 3D content. Again, the same conclusion can be drawn for different sizes of displays.

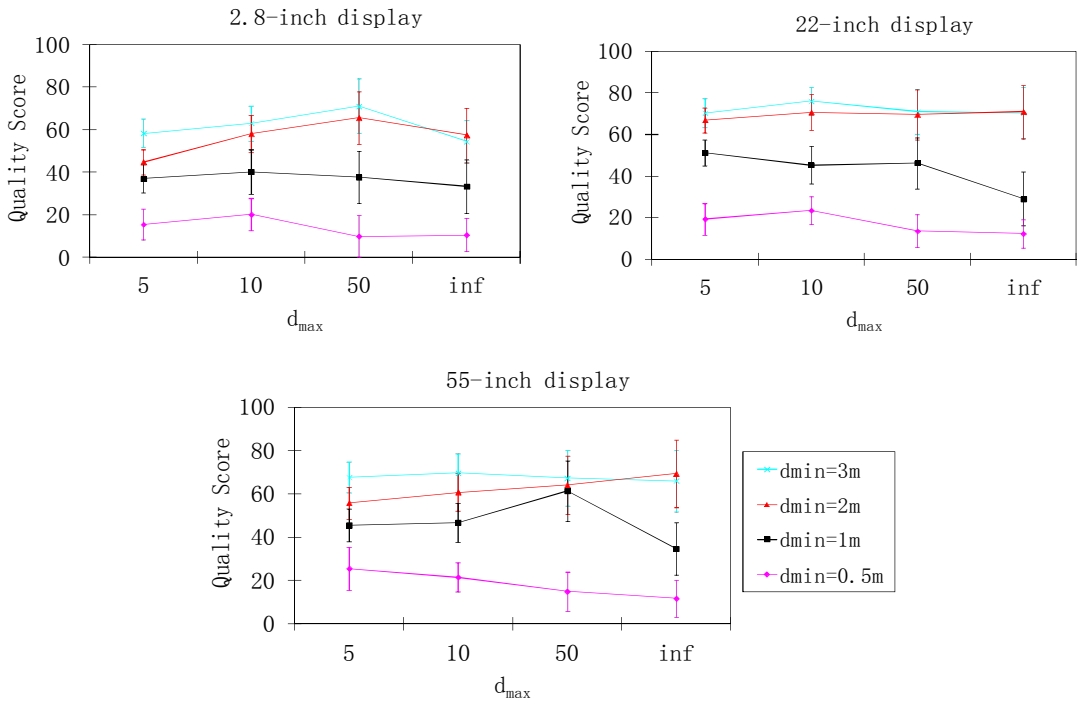


Figure 2.7: The mean opinion scores and their confidence intervals with various d_{max} (that is, 5m, 10m, 50m, and infinity) at different d_{min} on three sizes of displays.

2.3.3.4 Influence of d_{obj} not being the foreground object to image quality

We tested a few images where the object of interest is not the closest object in the image, that is, when d_{obj} is greater than d_{min}. We compared image sets with the same d_{obj} and the same d_{max}, but various d_{min}. The mean opinion scores and confidence intervals associated with different sizes of displays are shown in Figure 2.8. In each group of images, the left most bar is associated with the image where d_{min} equals d_{obj}, and the other bars are associated with images where d_{min} is less than d_{obj}. Having compared the four sets of images, we observe that the quality of most images is impacted to certain extent when some foreground objects, such as floor and ceiling, appear closer to the cameras than the object of interest.

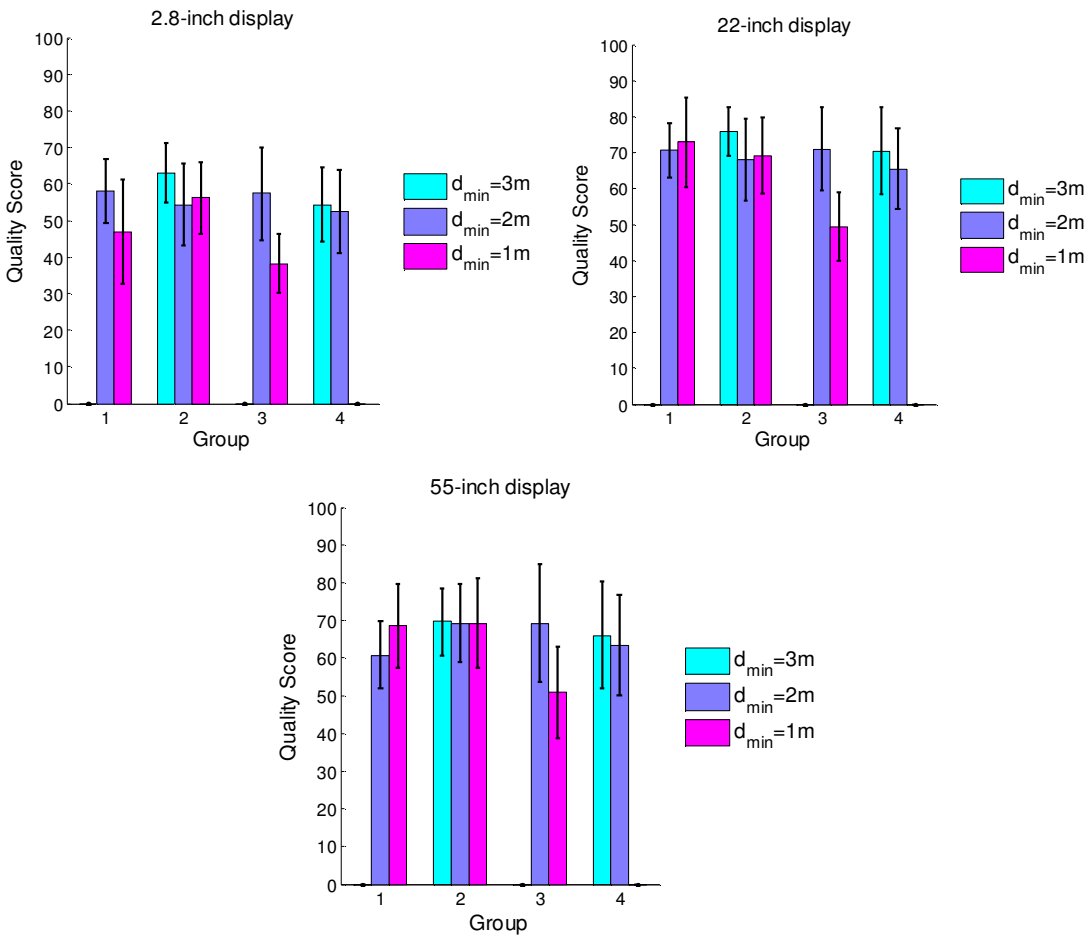


Figure 2.8: Comparison of the mean opinion scores and confidence intervals for four groups of content. Each group of content was captured at the same d_{max} and d_{obj} with different d_{min}. Group 1 was captured at d_{max}=10m and d_{obj}=2m, group 2 was captured at d_{max}=10m and d_{obj}=3m, group 3 was captured at d_{max}=infinity and d_{obj}=2m, and group 4 was captured at d_{max}=infinity and d_{obj}=3m.

2.3.4 Stage Two: Influence of Capturing Parameters to 3D Video Quality

Having discussed the influence of capturing parameters to 3D image quality, we study the influence of the same parameters to 3D video quality in the second stage of our experiment. All 3D videos in our database are with moderate motion. The subjective test was performed on a 65" 3DTV.

2.3.4.1 *Influence of lighting condition to video quality*

The statistical results of video quality are compared between the indoor and outdoor videos with the same set of parameters. No significant difference is found between videos taken under these lighting conditions. The conclusion is consistent with that of stage one.

2.3.4.2 *Influence of d_{min} to video quality*

In this comparison, we chose videos where d_{obj} equals d_{min}. These videos were divided into four groups according to the value of d_{max}. The subjective quality between videos taken at different d_{min} is compared in each group. Figure 2.9 shows the mean opinion scores and confidence intervals versus d_{min} at different d_{max} distances. For the same d_{max}, the video quality increases with d_{min} and levels off when d_{min} is greater than two meters. The confidence intervals when d_{min} is 0.5 meters are generally smaller than those with large d_{min}. This reflects the consistency of the observers' opinions on the low quality videos, where the closest objects were too close to the cameras. The same trend was found in stage one of our test.

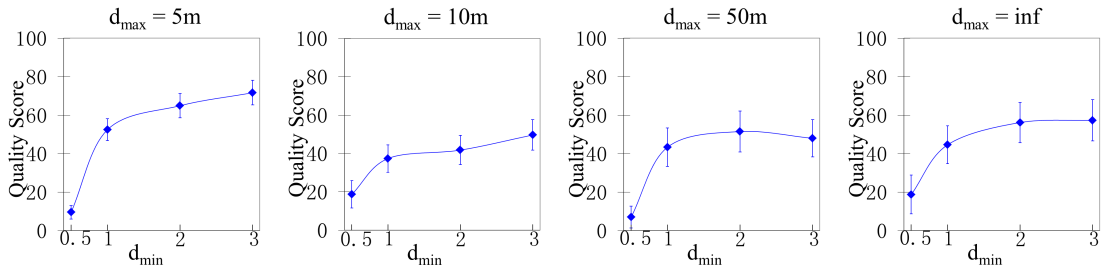


Figure 2.9: The mean opinion scores and their confidence intervals at various d_{min} (that is, 0.5m, 1m, 2m, and 3m). The four subplots correspond to the cases when d_{max} are 5m, 10m, 50m, and infinity.

2.3.4.3 *Influence of d_{max} to video quality*

We examine the influence of d_{max} to the video quality scores. Figure 2.10 shows the mean opinion scores of videos with different d_{max} while keeping the same d_{min}. No clear trend is found based on Figure 2.10, although the quality scores vary with d_{max}. The result is also consistent with that from stage one.

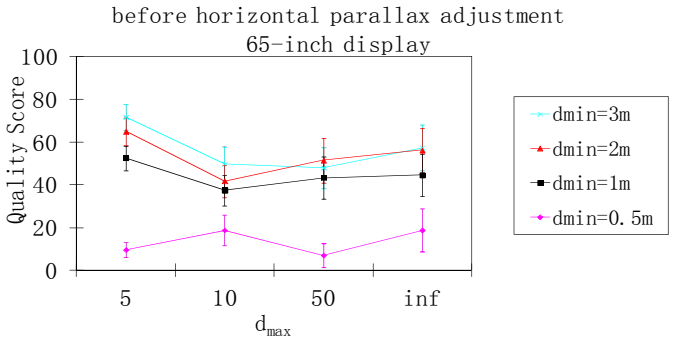


Figure 2.10: Comparison of the mean opinion scores at d_{max} equal to 5m, 10m, 50m, and infinity, with different d_{min}.

2.3.4.4 *Influence of d_{obj} not being the foreground object to video quality*

We chose three groups of videos. Within each group, the videos are captured with the same d_{obj}, the same d_{max}, and different d_{min}. Sample frames of some of these videos are shown in Figure 2.11. The mean opinion scores and confidence intervals of each group are shown in Figure 2.12. Based on the ratings of videos in group 3, we note that the grass being the foreground is annoying to the observers. The patterned floor in groups 1 and 2, however, increases 3D quality. Most observers prefer to watch videos with the patterned floor popping out of the screen and appearing in front of the object of interest.


Group 1: Left: d_{max}=10m, d_{obj}=2m, d_{min}=2m; Right: d_{max}=10m, d_{obj}=2m, d_{min}=1m.



Group 2: Left: d_{max}=10m, d_{obj}=3m, d_{min}=3m; Middle: d_{max}=10m, d_{obj}=3m, d_{min}=2m; Right: d_{max}=10m, d_{obj}=3m, d_{min}=1m.


Group 3: Left: d_{max}=infinity, d_{obj}=2m, d_{min}=2m; Right: d_{max}=infinity, d_{obj}=2m, d_{min}=1m.

Figure 2.11: Frames from three groups of videos used to examine the influence of d_{obj} not being the foreground object.

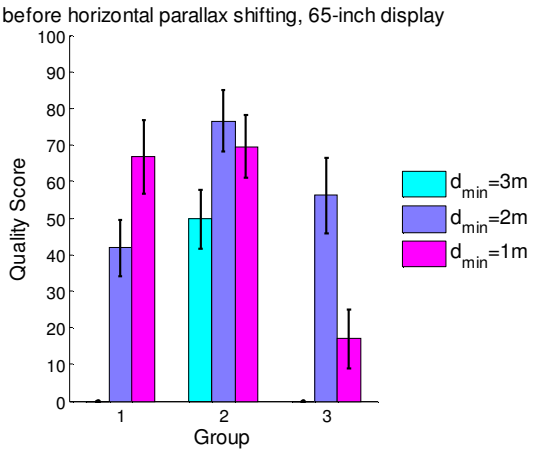


Figure 2.12: Comparison of the mean opinion scores and confidence intervals for three groups of videos before horizontal parallax adjustment on the 65-inch display. Each group of content was captured at the same d_{max} and d_{obj} with different d_{min}. Group 1 was captured at d_{max}=10m and d_{obj}=2m, group 2 was captured at d_{max}=10m and d_{obj}=3m, and group 3 was captured at d_{max}=infinity and d_{obj}=2m.

Since the results presented in Stage One and Stage Two are mainly consistent, we conclude that the influence of capturing parameters to 3D image quality and video quality are the same, providing no very fast motion is included in the video. Therefore, in Stage Three, we will focus only on 3D videos.

2.3.5 Stage Three: Influence of Capturing Parameters to 3D Video Quality after Horizontal Parallax Adjustment

The horizontal parallax adjustment was proposed to improve the 3D quality using simple post processing. In this section, we reveal how the parallax adjustment described in Section 2.1.4 affects the quality of 3D content.

2.3.5.1 Influence of lighting condition to video quality after horizontal parallax adjustment

Despite the quality changes brought by the horizontal parallax shifting, the influence of lighting conditions to 3D quality remains the same. In other words, there is still no significant quality difference between indoor scene with uniform artificial lighting and outdoor scene under direct sunlight.

2.3.5.2 Influence of d_{min} to video quality after horizontal parallax adjustment

The quality of 3D content after parallax adjustment is again significantly affected by d_{min}. Figure 2.13 shows the quality scores for videos with various d_{min} before and after horizontal parallax adjustment. The quality increases with d_{min} and it levels off at d_{min} equals 2m for videos both before parallax adjustment and after. The quality improvement by parallax adjustment is significant, with an exception of one video where d_{max}=5m and d_{min}=d_{obj}=3m. This exception is due to the fact that in this specific setup the depth bracket

(i.e., the amount of 3D space used in a shot or a sequence) is very small hence there is little room for quality improvement by parallax adjustment.

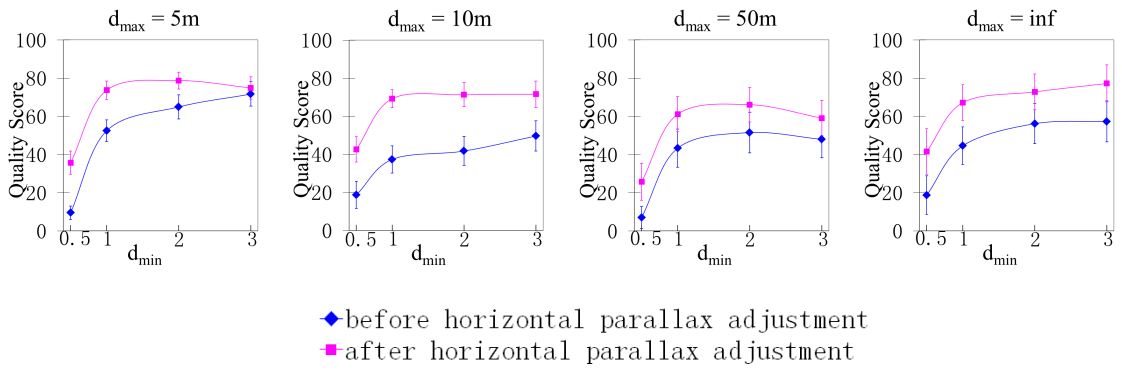


Figure 2.13: The mean opinion scores and their confidence intervals at various d_{min} (that is, 0.5m, 1m, 2m, and 3m). In reading order, the four subplots correspond to the cases when d_{max} are 5m, 10m, 50m, and infinity. The blue lines represent data before horizontal parallax adjustment and the pink lines show the data after horizontal parallax adjustment.

We can also conclude that although the horizontal parallax adjustment has greatly improved the quality of 3D content, very small d_{min} still leads to unsatisfactory quality (i.e., the mean opinion score is below 50 on a 0 to 100 rating scale in all four subplots in Figure 2.13), and hence, needs to be avoided in the capturing process.

We computed the mean opinion scores of all 25 test videos, that d_{min} equals d_{obj}, before and after horizontal parallax adjustment. The scores of all 25 video sequences indicate that viewers perceive horizontally adjusted videos to possess higher quality than non-adjusted ones, with an average quality-score gain of 19.86%.

2.3.5.3 Influence of d_{max} to video quality after horizontal parallax adjustment

Figure 2.14 shows the influence of d_{max} to the quality of 3D after horizontal parallax adjustment. It is interesting to know that the shape of Figure 2.14 is very similar to Figure 2.10, except that the quality level of each curve is raised.

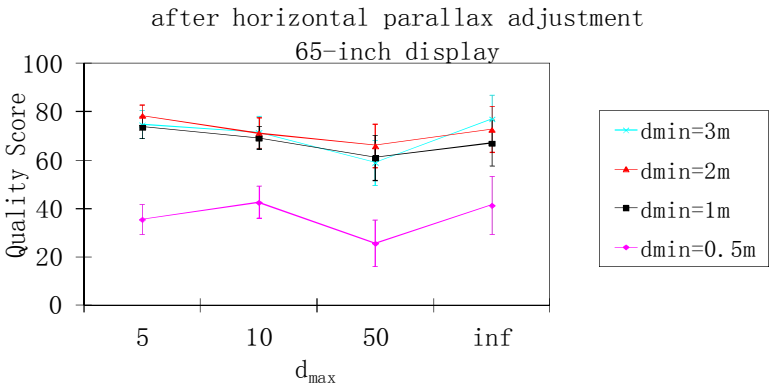


Figure 2.14: Comparison of the mean opinion scores at d_{max} equal to 5m, 10m, 50m, and infinity, with different d_{min}.

2.3.5.4 Influence of d_{obj} not being the foreground object to video quality after horizontal parallax adjustment

We tested the same three groups of videos that are shown in Figure 2.11 where the closest object in the scene is not the object of interest. The quality score of each group of videos after horizontal parallax adjustment are shown in Figure 2.15. Having compared the three groups of videos, we observe that the video quality is affected to certain extent when some foreground objects appear closer to the cameras than the object of interest.

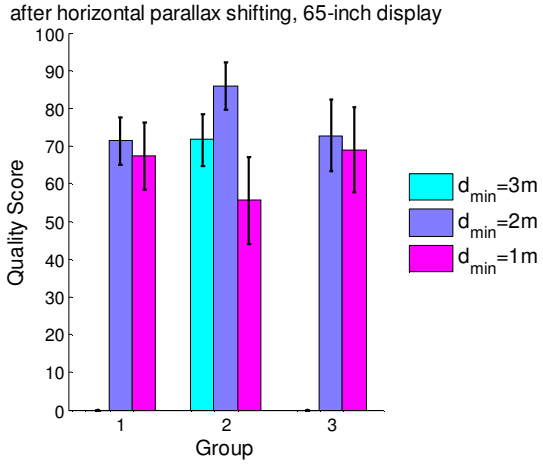


Figure 2.15: Comparison of the mean opinion scores and confidence intervals for three groups of videos after horizontal parallax adjustment. Each group of content was captured at the same d_{max} and d_{obj} with different d_{min}. Group 1 was captured at d_{max}=10m and d_{obj}=2m, group 2 was captured at d_{max}=10m and d_{obj}=3m, and group 3 was captured at d_{max}=infinity and d_{obj}=2m.

The second video in group 2 receives higher quality score because its content is hilarious and viewers tend to enjoy it more.

2.3.5.5 A 3D quality model based on capturing parameters

In this section, we derive a quality model for 3D videos. We only consider videos with adjusted horizontal parallax, since this simple adjustment significantly improves the video quality, and therefore is highly recommended.

Among the capturing parameters we considered, d_{min} and d_{max} are the most influential to 3D quality. In order to determine the effect of d_{min} and d_{max} on 3D quality, we consider the geometry of the stereoscopic imaging process shown in Figure 2.16. In this figure, the green tree is an object in the real world, and the red and blue trees are the images of the tree in the left and right cameras, respectively. The grey tree shows the position of the blue tree when the left image is superimposed on the right image. The parameter u is the

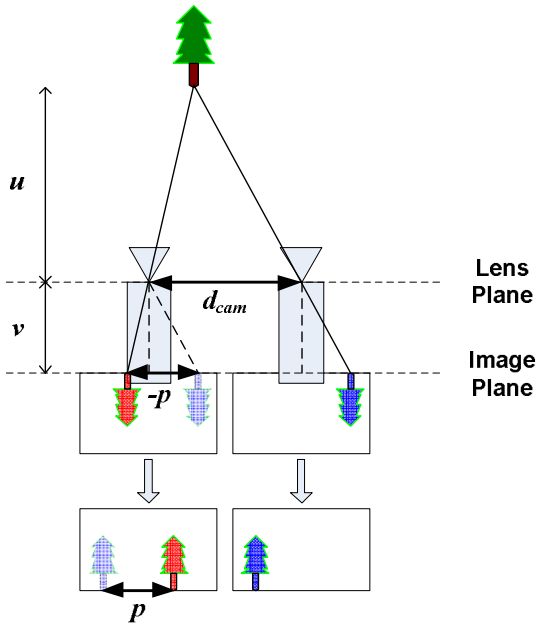


Figure 2.16: Geometry of the stereoscopic imaging process.

lens-to-object distance, v is the lens-to-image-plane distance, d_{cam} is the distance between the two cameras, and p is the horizontal parallax between the same object on the right and left images.

Based on Figure 2.16, we have

$$\frac{d_{cam}}{u} = \frac{-p}{v} \tag{2.1}$$

Furthermore, the relation among the imaging parameters can be described as

$$\frac{1}{f} = \frac{1}{u} + \frac{1}{v} \tag{2.2}$$

where f is the focal length of the camera. Based on (2.1) and (2.2), the horizontal parallax p can be rewritten in the following form:

$$p = -\frac{d_{cam}}{u} \cdot v = -\frac{d_{cam}}{u} \cdot \frac{u \cdot f}{u - f} = -\frac{d_{cam} \cdot f}{u - f} \tag{2.3}$$

Note that p is negative for a parallel camera setup. When $u >> f$, equation (2.3) becomes $p = -\frac{d_{cam} \cdot f}{u - f} \approx -\frac{d_{cam} \cdot f}{u}$. Hence, the range of the horizontal parallax of a stereoscopic image pair is $\left[-\frac{d_{cam} \cdot f}{d_{min}}, -\frac{d_{cam} \cdot f}{d_{max}}\right]$. Let s and r denote the sensor width and image horizontal resolution. Then the range of horizontal pixel parallax can be represented as $\left[-\frac{d_{cam} \cdot f \cdot r}{d_{min} \cdot s}, -\frac{d_{cam} \cdot f \cdot r}{d_{max} \cdot s}\right]$.

After horizontal parallax adjustment, all parallaxes are changed by $\frac{d_{cam}\cdot f\cdot r}{d_{min}\cdot s}$ pixels, making the smallest negative parallax becoming zero, and all other negative parallaxes becoming positive. The new range of horizontal pixel parallax after image interpolation (for keeping the original resolution) is:

$$\left[0,\ d_{cam}\cdot f\cdot\frac{r}{s}\cdot\left(\frac{1}{d_{min}}-\frac{1}{d_{max}}\right)\cdot\frac{r}{r-\frac{d_{cam}\cdot f\cdot r}{d_{min}\cdot s}}\right]=\left[0,\ d_{cam}\cdot f\cdot r\cdot\left(\frac{1}{d_{min}}-\frac{1}{d_{max}}\right)\cdot\frac{d_{min}}{d_{min}\cdot s-d_{cam}\cdot f}\right] \quad (2.4)$$

The 3D video quality after horizontal parallax adjustment is a function of the maximum pixel parallax *x*.

Therefore, we model the subjective test quality scores q as a function of x. That is, $q=F(x)$, where $x=d_{cam}\cdot f\cdot r\cdot\left(\frac{1}{d_{min}}-\frac{1}{d_{max}}\right)\cdot\frac{d_{min}}{d_{min}\cdot s-d_{cam}\cdot f}$. If we substitute the parameters of the stereoscopic cameras, then x becomes:

$$x=0.077\cdot 0.043\cdot 1920\cdot\left(\frac{1}{d_{min}}-\frac{1}{d_{max}}\right)\cdot\frac{d_{min}}{d_{min}\cdot 0.036-0.077\cdot 0.043} \quad (2.5)$$

which can be simplified as

$$x=6.3571\cdot\left(\frac{1}{d_{min}}-\frac{1}{d_{max}}\right)\cdot\frac{d_{min}}{d_{min}\cdot 0.036-0.003311} \quad (2.6)$$

We further perform a curve fitting by plugging in various capturing parameters (d_{min} and d_{max}) and their corresponding quality scores q to the equation $q=F(x)$. Then, the function $F(x)$ is approximated by a second order polynomial as

$$q = F(x) = -0.0003x^2 + 0.0315x + 72.2322 \quad (2.7)$$

Since x is a function of d_{min} and d_{max}, the 3D quality q can also be represented as a function of d_{min} and d_{max}, which is shown in Figure 2.17. From the figure, we observe that a small d_{min} significantly reduces the 3D quality, while a small d_{max} increases the 3D quality to some extent. The effect of d_{max} is negligible when d_{min} is large.

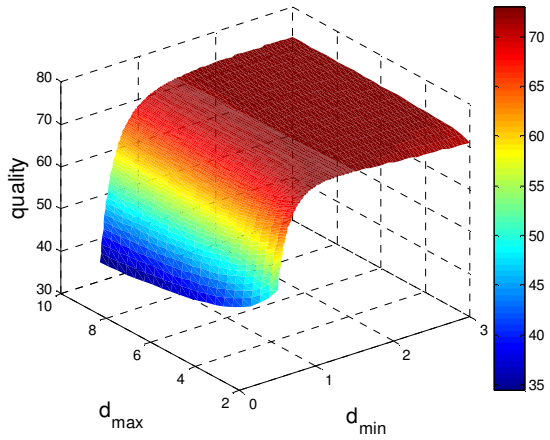


Figure 2.17: A function of quality in terms of d_{min} and d_{max}.

Although the above model is based on the specific camera setup used in our capturing process, the relationship holds in general. To obtain a pleasant 3D viewing experience, the above model can be considered as a useful guideline at the stereoscopic content capturing stage. Based on the desired quality, certain capturing parameter sets should be avoided.

2.4 Conclusions

We conducted comprehensive subjective tests to determine the influence of a few capturing parameters (i.e., lighting conditions, d_{obj}, d_{max}, and d_{obj} not being the foreground object) on the quality of 3D images, quality of 3D videos, and quality of 3D videos after

horizontal parallax adjustment when viewed on 3D TVs and 3D mobile devices. The influences of these parameters are consistent over different sizes of displays and over images and videos before and after horizontal parallax adjustment. The parameter d_{min} is the main factor affecting the 3D quality. Having d_{obj} (object of interest) not being the foreground object slightly degrades the 3D quality, whereas the lighting conditions (indoors and outdoors) and d_{max} do not have a significant effect on 3D quality. The automatic horizontal parallax adjustment algorithm that we implemented has greatly improved the 3D quality of experience by 19.86%. Despite such quality improvement, very small d_{min} still leads to poor quality and hence needs to be avoided in the capturing process.

3 Smart Stereoscopic 3D Video Reframing

As described in the introduction, the 2D VAM and reframing problem have been well studied (Section 1.3.2). In comparison the body of work on 3D VAM and reframing is much more limited.

In this chapter, we propose an automatic 3D video reframing algorithm based on a novel 3D VAM. Unlike previously proposed 2D and 3D VAMs, our 3D VAM considers depth as well as other saliency indicators such as luminance, color, and motion to identify prominent regions of stereoscopic 3D content. The model is proven to be effective for a great variety of 3D videos that are commonly encountered in real life as opposed to very few images/videos captured in a simple lab environment. This 3D VAM is robust, cost efficient, and suitable for real-time implementation. Additionally, our 3D reframing method provides a dynamic bounding box that slides smoothly from frame to frame and keeps the visually important regions within the box. The smooth temporal transition of the bounding boxes is again achieved by computational efficient steps.

The rest of this chapter is divided as follows. Section 3.1 presents our novel 3D reframing algorithm, including the proposed 3D visual attention model, the choice of the bounding box, and the temporal smoothing process. Section 3.2 shows our experimental results and analysis. The subjective evaluations are discussed in Section 3.3. Finally, conclusions are drawn in Section 3.4.

3.1 Proposed Automatic 3D Video Reframing Algorithm

Figure 3.1 shows the block diagram of our proposed automatic reframing algorithm. As it can be seen, our approach consists of a 3D visual attention model and a smart reframing algorithm with smooth transition. The 3D VAM computes luminance, disparity, motion, and color to generate a local edge saliency map, a disparity saliency

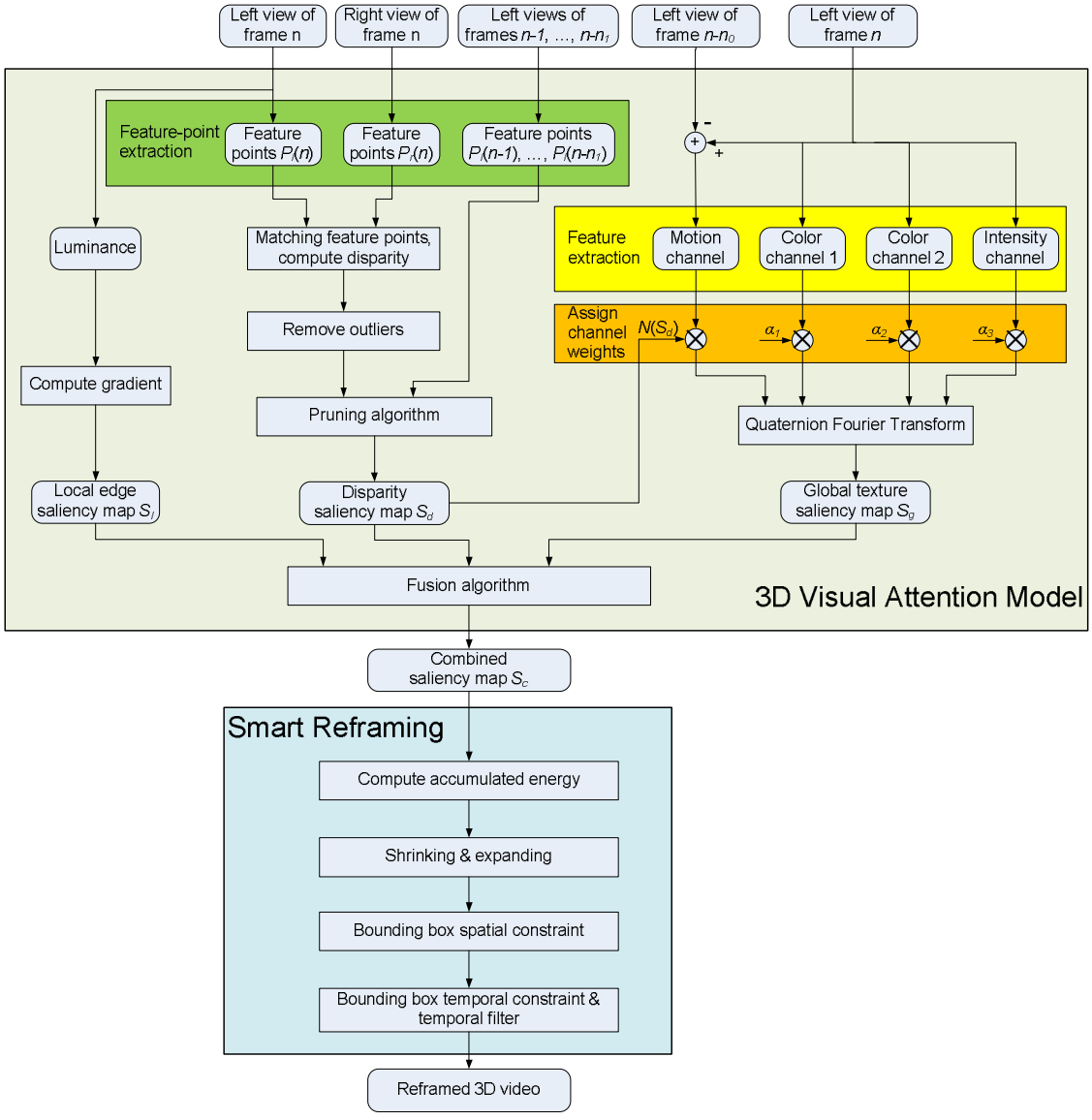


Figure 3.1. Block Diagram of the proposed reframing algorithm.

map, and a global texture saliency map. These three maps are then fused into one to identify the salient areas of a 3D scene. The smooth transition of bounding boxes is achieved by applying a few constraints and filtering processes, as depicted in Figure 3.1. In the following subsections, we describe our 3D visual attention model and reframing process.

3.1.1 3D Visual Attention Model

We have developed a visual attention model for stereoscopic 3D content that combines data obtained from three different types of saliency maps, namely, the local edge saliency map S_l, the disparity saliency map S_d, and the global texture saliency map S_g. The final combined saliency map S_c is obtained using a weighted average of the three maps:

$$S_c = \beta_l N(S_l) + \beta_d N(S_d) + \beta_g N(S_g) \tag{3.1}$$

where each β is a scalar, and $\beta_l + \beta_d + \beta_g = 1$.

Local edges emphasize the boundary of the objects contained in each video frame. By obtaining the edges of these objects, we create a "line drawing" of the scene that emphasizes regions with changing surfaces. This information is useful since "busy" regions tend to be appealing to the human eye.

Disparity-based saliency assumes that objects that are close to the camera draw more visual attention than distant objects. This information can be obtained by comparing the left and right views of a stereoscopic frame.

In addition, global texture saliency refers to basic visual features that attract people's attention such as color, spatial frequency, brightness, and motion. In particular, the combination of motion and depth is crucial for identifying the main visual regions of a video frame.

Figure 3.2 provides an example of how our scheme employs these three basic maps to produce a definitive saliency map for 3D video. Figure 3.2(a) shows the left view of the original side-by-side 16:9 3D frame. The local edge saliency map, disparity saliency map, and global texture saliency map are respectively shown in Figure 3.2(b), (c), and (d). The combined saliency map is given in Figure 3.2(e). The map suggests that the most salient region is the person, which is the only section of the frame with significant movement

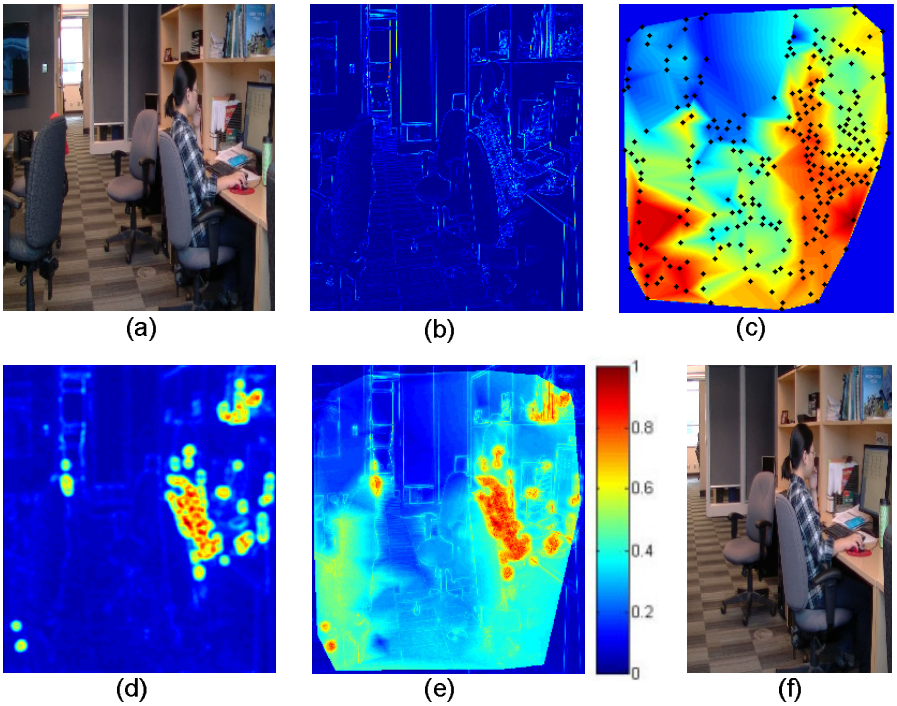


Figure 3.2. An example of the proposed 3D visual attention model and its saliency maps. For illustration purposes, we only show the left frame. (a) Original frame (horizontally squeezed since it is the left half of the side-by-side 16:9 3D frame); (b) local edge saliency map s_l; (c) disparity saliency map s_d; (d) global texture saliency map s_g; (e) combined saliency map s_c based on all three maps; (f) resulting cropped frame with a 4:3 aspect ratio.

(the sequence is handheld so there is relative movement in the entire frame). The colorbar beside Figure 3.2(e) indicates the color and its corresponding saliency value for all normalized saliency maps, that is, Figure 3.2(b), (c), (d), and (e).

In the following three subsections, we describe the methods for computing the three saliency maps in detail.

3.1.1.1 Local Edge Saliency Map

We compute a local edge saliency map for each frame. For simplicity, this map is obtained by computing the gradient of the frame's luminance component *L*. The magnitude of this gradient is then chosen as the local edge saliency map S_l., as shown in the following equation:

$$S_l = \|\nabla L\|. \tag{3.2}$$

An example is shown in Figure 3.2(b).

3.1.1.2 Disparity Saliency Map

People tend to give more importance to objects that are closer to them than to the ones that are further back. Information about the closeness of objects can be obtained by comparing the left and right views of each frame. In what follows, we show the steps for computing a disparity-based saliency map.

In this work, we aim at developing a real-time reframing solution. In order to retain a fast algorithm, we first downsample the left and right views of each stereoscopic frame by a factor of 2. We then extract and match distinctive feature points between the downsampled views employing a very fast shift-invariant feature matching algorithm [56]

[57]. Disparities of each pair of matching points are computed. Next, we remove some pairs of the matching points to further ensure the matching accuracy of the remaining points. This is done by discarding the points with large vertical disparities and points with horizontal disparities that heavily deviate from the majority of the points. A pruning algorithm [33] is then used to retain a set of sparse feature points in order to further increase the robustness and accuracy. This pruning algorithm trims the less robust points based on their temporal stability. To this end, we match and track all feature points among the neighboring $2n_1 + 1$ frames, i.e., frames $n - n_1$ to $n + n_1$. The feature points

are then sorted according to their repeat time, which indicates the stability of the feature points over time. Next, the pruning algorithm removes low stability points that are near a high stability point if there are little disparity differences between the low and high stability points. This process spatially prunes clustered points, while preserving points that are associated with objects of different disparities. In detail, a greedy algorithm is used. Let $p(x, y)$ be an unprocessed feature point with the highest stability, and $p'(x', y')$ be any other feature point in the same frame. From the set of feature points, we remove all points p' that satisfy

$$\left\| \begin{pmatrix} \mathrm{x} \\ \mathrm{y} \\ \mathrm{d_x(x,y)} \\ \mathrm{d_y(x,y)} \end{pmatrix} - \begin{pmatrix} \mathrm{x'} \\ \mathrm{y'} \\ \mathrm{d_x(x',y')} \\ \mathrm{d_y(x',y')} \end{pmatrix} \right\| < r, \tag{3.3}$$

where $d_x(x, y)$ and $d_y(x, y)$ are the horizontal and vertical disparities, respectively, at point $p(x, y)$, and r is an isotropic distance threshold measured in spatial and disparity spaces. The pruning process is repeated until all feature points in a frame are processed.

A point is removed if it is temporally less robust and its disparity is similar to a more robust neighboring point.

Subsequently, a dense disparity map (one value per pixel) is generated by linearly interpolating the sparse feature points based on Delaunay triangulation [58]. Areas near the frame boundary are usually not included in any Delaunay triangles. These regions do not have any interesting points, and are not the region of interest. Therefore, we assign the maximum disparity value to them. See the dark blue area in Figure 3.2(c) as an example. The disparity-based saliency map is finally obtained by assigning high saliency to small disparity values and low saliency to large disparity values. Figure 3.2(c) shows an example of this type of map. The robust feature points are superimposed on the disparity saliency map S_d.

3.1.1.3 Global Texture Saliency Map

It has been reported [59] that viewers pay special attention to basic visual features such as color, brightness, and motion. Therefore, it is important to use this information to determine the salient objects of video frames. Although several computational models have been proposed to simulate human visual attention [60], we decided to use the scheme proposed in [61] as a starting point for our own global texture saliency map since it is fast and produces better results than other state-of-the-art schemes. As in [61], we also use the Quaternion Fourier Transform (QFT) [62] to produce a saliency map. An important difference, however, is that we adaptively assign different weights to the four channels, and we use the 3D disparity saliency map to weigh the motion channel. This

allows us to control which of the features will have a stronger impact on the global texture saliency map and the final VAM.

We represent each pixel with a special type of complex number called a *quaternion q* [62] which includes four terms:

$$q = a + bi + cj + dk. \tag{3.4}$$

The terms a, b, c, d are real numbers associated with the four channels. The complex operators i, j and k are orthogonal to each other. The quaternion can also be written as a complex number whose "real" and "imaginary" components are themselves complex numbers:

$$q = A + Bj \tag{3.5}$$

where $A = a + bi$ and $B = c + di$.

For every pixel, we express information related to color, intensity, and motion in the form of a quaternion. This allows us to obtain a quaternion video frame. Each quaternion video frame is composed of two color channels, an intensity channel, and a motion channel. The color channels C_1 and C_2, are, respectively, red/green and blue/yellow following the 'color opponent-component' system introduced in [63]. The intensity channel, I, is the average value of the R, G and B components. Each of these three channels (two colors and intensity) is normalized to one.

In our algorithm, we propose the following modifications to the motion channel for fast implementation and improved performance compared with [61]. To compute the

motion channel M, we first compute the absolute difference between the intensity values of the current frame and a previous frame, as follows:

$$\Delta I = |I(n) - I(n - n_0)|, \tag{3.6}$$

where $I(n)$ is the intensity of the current frame n, $I(n - n_0)$ is the intensity of a previous frame, and n_0 is a small positive integer, which will be determined based on experiments over many test sequences. The empirical value will be determined later in Section 3.2. Then, this absolute difference ΔI is normalized so that the highest value equals 1. For every pixel, the motion channel M is defined as:

$$M = \begin{cases} N(\Delta I), & \text{when } N(\Delta I) > \tau_M, \\ 0, & \text{otherwise,} \end{cases} \tag{3.7}$$

where N is a normalization operator, and τ_M is a threshold that satisfies $0 < \tau_M \leq 1$. The purpose of setting some values to zero is to eliminate all the information related to small movements or slight brightness changes that the video might have, and only focus on the significant motion information.

The quaternion frame q in our proposed method is represented as follows:

$$q = N(S_d)M + \alpha_1 C_1 \mu_1 + \alpha_2 C_2 \mu_2 + \alpha_3 I \mu_3, \tag{3.8}$$

where μ_i, $i = 1, 2, 3$ satisfies $\mu_i^2 = -1$, $\mu_1 \perp \mu_2$, $\mu_2 \perp \mu_3$, $\mu_3 \perp \mu_1$, $\mu_3 = \mu_1 \mu_2$; N is the normalization operator; α_1, α_2, and α_3 are constant values between 0 and 1.

Compared to directly taking the four channels (i.e., M, C_1, C_2, and I) as the quaternion terms $q = M + C_1 \mu_1 + C_2 \mu_2 + I \mu_3$ as in [61], we have added weights to each of the four

channels as shown in equation (3.8). This decision was taken after implementing the original scheme and conducting several subjective tests with a small group of people. We have determined that, for 3D videos, the motion channel is more relevant than the other three channels to create an effective global saliency map. Therefore, we highlight the motion channel M with respect to the other three. We use the normalized disparity map $N(S_d)$ as the weighting factor for channel M, since for the same amount of movement, the motion associated with a foreground object is perceived to be more important than that associated with a background object. By incorporating the 3D disparity information, we are able to obtain an effective 3D global texture model.

We then apply the Quaternion Fourier Transform (QFT) to every quaternion frame q and obtain the transformed signal Q. The QFT is a special frequency transformation that treats a color image as a vector field (the quaternion). The phase spectrum of the QFT specifies where each of the sinusoidal components resides within the image. Locations with less periodicity (more texture) stand out and are recognized as the map's salient points. We implemented this method using the quaternion toolbox for Matlab available in [64].

In order to obtain the global texture saliency map S_g for each frame, we represent Q in polar form and normalize the magnitude of each of its elements to 1. The resulting function, Q' only contains the phase information of Q. We then apply the inverse QFT to Q' to obtain q'. The saliency map is obtained by filtering the magnitude of q' with a Gaussian filter. Finally, we dilate the resulting map to obtain significant clustered regions.

An example of a global texture saliency map S_g is shown in Figure 3.2(d). The map indicates that the most salient regions of the frame are the person, the books, and the computer screen.

3.1.2 Automatic 3D Video Reframing with Smooth Transition

Our automatic stereoscopic 3D video reframing solution produces the three saliency maps described in Section 3.1.1 and fuses them to create a single model for visual attention. There are several proposals for combining saliency maps such as the schemes detailed in [47]. In our method, the maps are normalized and averaged to obtain the combined saliency map, which is later shown to be very effective for 3D reframing

For the case of video, decisions on how to crop a frame so that it fits the new aspect ratio cannot be solely based on the information available from its associated saliency map. We also need to consider the cropping locations of the previous frames so that we can ensure that the location of the bounding box does not change abruptly, producing a shaky video. In order to do this, we have designed a scheme that provides a smooth temporal transition bounding box.

The first goal of our scheme is to identify the area in the saliency map with the highest "energy." The energy in an area is defined as the summation of all saliency values within this area. For fast implementation, an accumulated energy matrix is pre-computed. We normalize the accumulated energy matrix so that the maximum value in the matrix is 1. The value of this matrix at each location P, denoted as $E(\mathrm{P})$, is calculated as the energy of the rectangular region defined between the pixel on the top-left corner of the map and the current pixel P. Then, the energy in any rectangular region in the map can be later

computed as three summations rather than requiring the sum of all the pixel values in this area. As shown in Figure 3.3, the energy in the rectangle $ABCD$ can be simply computed as:

$$E(ABCD) = E(A) - E(B) - E(D) + E(C). \quad (3.9)$$

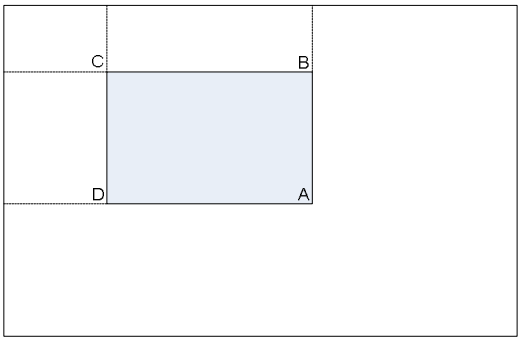


Figure 3.3. The energy in the rectangle ABCD is defined as *E(ABCD) = E(A) - E(B) - E(D) + E(C).*

Based on the desired aspect ratio, we crop the frame leaving the rectangular region that contains the highest energy.

Quite often, parts of an object have high saliency values whereas other parts have low values. Reframing solely based on the energy of a saliency map may result in cropping some important object. In order to avoid this, we propose to use a very simple yet effective approach. First, we reduce the size of the bounding box by *w* pixels when searching for the highest energy area. The value *w* – which is based on video resolution - is empirically determined later on (in Section 3.2) through extensive experiments. . Next, we expand the bounding box by *w* pixels on all sides with the purpose of including the entire important object in the cropped new frame. This shrinking and expanding approach also implicitly brings the salient area towards the center of the new frame. Furthermore,

this scheme reduces the probability of experiencing a window violation after reframing. Figure 3.4 illustrates the effect of applying the shrinking and expanding algorithm. Figure 3.4(a) shows the original frame. Figure 3.4(b) depicts a bounding box that contains the maximum energy superimposed on the saliency map, which is obtained using our algorithm described in Section 3.1.1. Figure 3.4(c) is the corresponding cropped frame, which leaves out a part of the girl on the right most side. This undesirable result is because the left bottom corner of the saliency map contains higher energy than the right side of the girl in white. On the contrary, the result after applying the shrinking and expanding algorithm is shown in Figure 3.4(d) and (e). The shrunk window is depicted

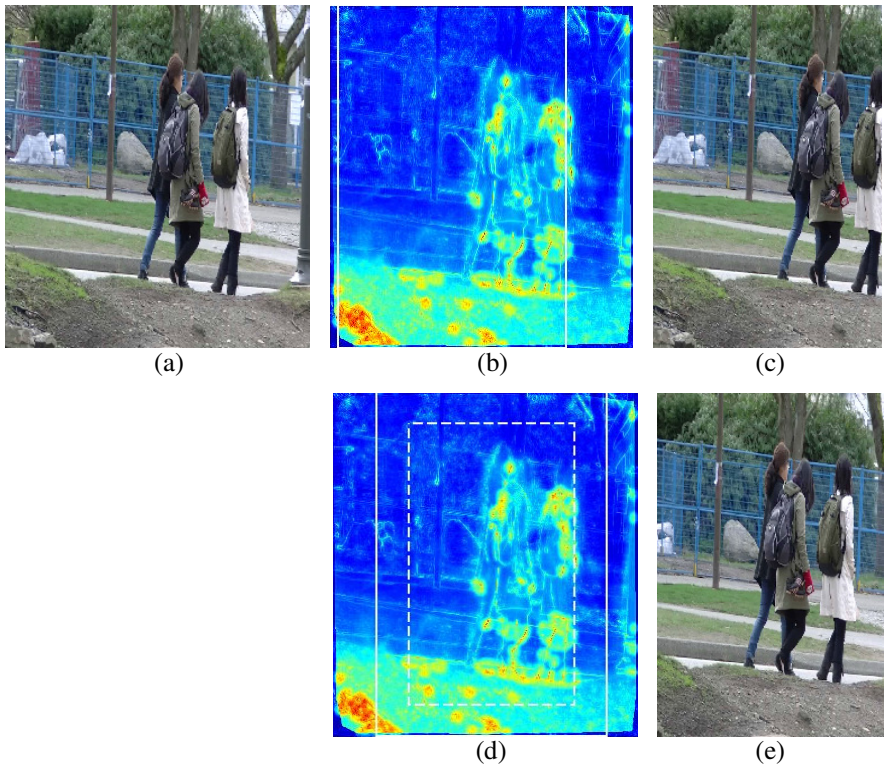


Figure 3.4. Illustration of the shrinking and expanding algorithm. (a) Original frame; (b) bounding box that contains the maximum energy superimposed on the combined saliency map; (c) the cropped frame associated with the bounding box in (b); (d) bounding box (solid line) obtained with our proposed shrinking and expanding algorithm superimposed on the combined saliency map; The dotted line is the shrunk bounding box that contains the maximum energy; (e) the cropped frame associated with the bounding box in (d).

with dotted lines, where contains the maximum energy of its size; whereas the expanded bounding box is shown in the solid lines, containing all three girls in the frame and bringing them towards the centre of the cropped frame. The success of this step brings the entire algorithm towards "understanding objects" while it eliminates the complex procedures introduced by steps such as "object segmentation".

In order to reduce the computational cost and ensure smooth temporal transition for the cropping window, we first make sure that the locations of the consecutive frames are spatially constrained if no scene change is detected. That is, the location difference of two consecutive frames is smaller than a threshold δ. The value of δ, which will be specified in Section 3.2, is determined by the resolution of the original video and the amount of motion generally occurred in a sequence.

Although a constraint of cropping locations is set in the previous step, local jerks still exist. This is often caused by small differences on the consecutive saliency maps which are resulted from insignificant motion or lighting changes. For this reason, when choosing the bounding box for the current frame, we give higher priority to the bounding box location of the previous frame. To this end, if the energy increase associated with the new location is less than a threshold τ_E, we keep using the previous location. The value of τ_E will be derived in Section 3.2 as a result of performance evaluations using many video sequences.

Further reduction of flickering may be achieved by keeping temporal variations below 0.5 Hz, which is based on the temporal frequency response of the human visual system [65]. To achieve this, we employ a 29 tap windowed linear-phase FIR lowpass

filter with a cutoff frequency of 0.5Hz to the bounding box location, to further ensure the smoothness of the cropping window over time. Finally, the cropping locations are rounded to integers after applying the temporal filter in order to avoid spatial interpolation of a frame. The effects of giving high priority to the prior location and using the temporal filter are illustrated in Figure 3.5. It shows the trajectory of the bounding box location for the sequence "Lounge". Figure 3.5(a) depicts the result without using the

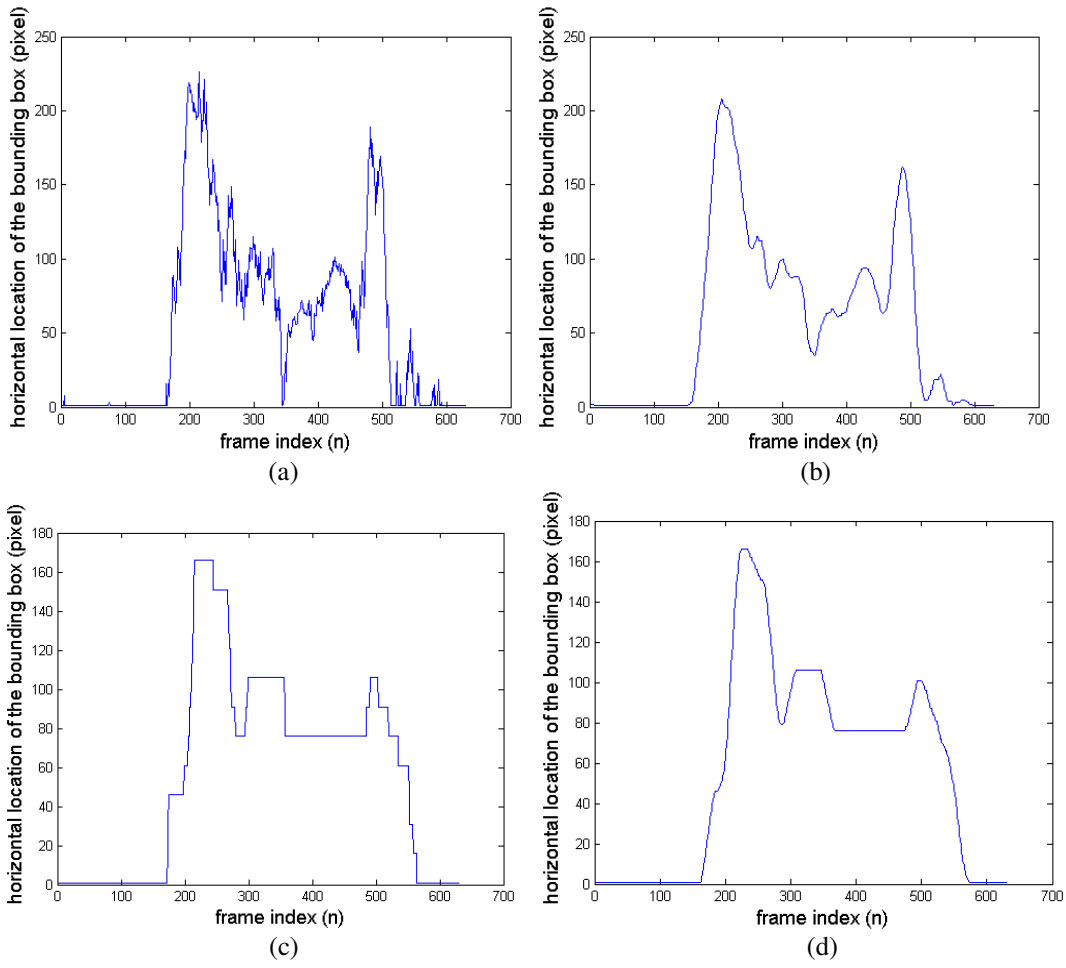


Figure 3.5. Illustration of the effects of giving high priority to the prior location and using temporal filter to the trajectory of the bounding box. Bounding box locations (a) without giving high priority to the prior location or using the temporal filter; (b) only using the temporal filter; (c) only giving high priority to the prior location; (d) giving high priority to the prior location and using the temporal filter.

prior location information or using the temporal filter. In this case, the trajectory is very noisy, thus resulting in major jittering. Figure 3.5(b) shows the result after only applying the temporal filter. The trajectory is smooth, however the bounding box moves back and forth too frequently, which is annoying and undesirable. The result of only giving high priority to the prior location is shown in Figure 3.5(c), where sudden location jumps lead to dramatic transitions. Figure 3.5(d) gives the trajectory when both the high priority is assigned to prior location and the temporal filter is applied. The smooth and relatively slow location change removes the jittering problem and provides pleasant reframed video sequences.

3.2 Experimental Results

We captured dozens of HD (high definition) stereoscopic video sequences using a JVC Everio GS-TD1 3D camcorder. Each video frame is composed of a side-by-side left and a right view, each with an 8:9 aspect ratio, resulting in a 3D frame with a 16:9 aspect ratio. This format is widely accepted by 3D displays of 16:9 aspect ratios. The resolution of the side-by-side frame is 1920 pixels × 1080 pixels.

We reframed these videos to a 4:3 aspect ratio (i.e., a 2:3 aspect ratio for each view) using the proposed method. In what follows, we present the values for the parameters used in the 3D reframing algorithm. These values are carefully determined through extensive experiments and provide effective and robust reframing results for a large variety of video sequences. We used a value of 5 for n_0, which means that the motion channel uses information from the current frame at time n and the frame at n - n_0. The threshold τ_M was set to 0.6 and the weights α_1, α_2, and α_3 were all set to 0.1. In our

experiment, we use $n_1 = 14$ in the pruning algorithm. In other words, we compute the stability of the feature points by tracking them over 29 (that is, $2n_1 + 1$) frames. We use $r = 10$ as the isotropic threshold of the pruning algorithm. The local-edge, disparity, and global-texture saliency maps play important roles in the process of generating a final map that accurately identifies the prominent regions of a 3D video frame. We empirically chose to employ $\beta_l = 1/3$, $\beta_d = 1/3$, and $\beta_g = 1/3$ in the fusion algorithm for generating a combined 3D saliency map. The proposed weights balance very effectively the effects of the three saliency maps.

For HD stereoscopic video sequences, we employ *w* equal to 100 pixels in the shrinking and expanding stage. We found in our experiments that a δ value of 15 pixels is able to sufficiently track the moving objects and maintain a relatively constrained position of the bounding box. The energy increase threshold τ_E is set to 1% to effectively avoid the location change caused by small energy variations in the consecutive frames.

The following five examples show how our 3D VAM computes and combines the three saliency maps into a single meaningful map for various video sequences. In Section 3.1.1, Figure 3.2 shows a frame from the sequence “Lab” and the various saliency maps obtained with our method. The disparity map (Figure 3.2(c)) emphasizes the objects that are closer to the cameras, mainly the chair at the left of the frame and the person working. On the other hand, the global texture map (Figure 3.2(d)) highlights three main regions: the person, the books, and the computer. Once the maps are put together as in Figure 3.2(e), most of the energy is concentrated on the person since she is prominent in both the disparity and the global texture maps.

Figure 3.6 shows the various saliency maps created for one of the frames of a sequence called “Playground”, which was captured with a handheld camera. In this example, both the disparity map (Figure 3.6(c)) and the global texture map (Figure 3.6(d)) indicate that the most relevant region of the frame is the little girl. The global texture map highlights her because she is moving, while she is emphasized by the disparity map because she is close to the 3D camera. The combined saliency map (Figure 3.6(e)) clearly indicates that the little girl occupies the most salient region of the frame. This map informs our reframing method where to place the bounding box.

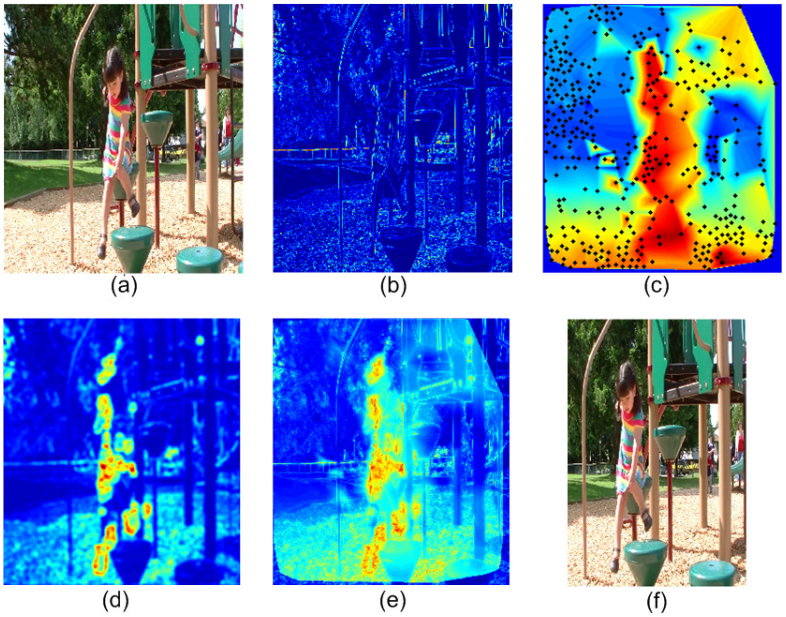


Figure 3.6. A frame from the sequence "Playground." (a) Original frame (horizontally squeezed since it is the left half of the side-by-side 16:9 3D frame); (b) local edge saliency map s_l; (c) disparity saliency map s_d; (d) global texture saliency map s_g; (e) combined saliency map s_c based on all three maps; (f) resulting cropped frame with a 4:3 aspect ratio. In this example, both the disparity saliency map and the global texture saliency map identify the little girl as the most salient region. The combined map informs our method where to place the bounding box.

Another example is illustrated in Figure 3.7, which includes a frame from the sequence “Main Mall,” captured with a 3D camera placed on a tripod. In this case, both the local edge map (Figure 3.7(b)) and the disparity map (Figure 3.7(c)) highlight the

bicycles. On the other hand, the global texture map (Figure 3.7(d)) highlights the people walking down the street as the main object on the frame. Finally, the combined map (Figure 3.7(e)) identifies all the main regions of the frame. This allows our method to select a bounding box for reframing purposes. Figure 3.7 is a clear example of the importance of employing a vast number of features to identify the objects of highest visual interest for each stereoscopic video frame. The combination of the various saliency maps provides an accurate visual attention model for 3D content.

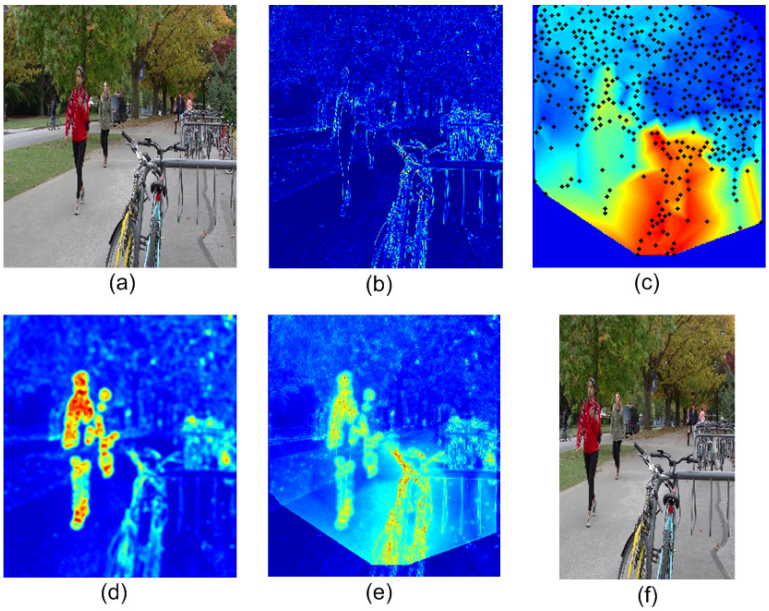


Figure 3.7. A frame from the "Main Mall" sequence. (a) Original frame (horizontally squeezed since it is the left half of the side-by-side 16:9 3D frame); (b) local edge saliency map s_l; (c) disparity saliency map s_d; (d) global texture saliency map s_g; (e) combined saliency map s_c based on all three maps; (f) resulting cropped frame with a 4:3 aspect ratio. In this case, the local edge map and the saliency map highlight the bicycle, which is closer to the 3D camera. However, the global texture map highlights the regions with motion (people walking). The combined map includes all the salient points in the frame and the bounding box is chosen accordingly.

Figure 3.8 shows a frame for the sequence “Black Truck.” This is a good example of a challenging video sequence. Here, both the local edge map (Figure 3.8(b)) and the global texture map (Figure 3.8(d)) fail to identify the person walking as a salient object. There are many reasons for this. While it is true that the person is moving, the camera is

also moving and tracking the person. Thus, all the pixels in the screen are changing positions on every frame, and the person remains relatively steady. Therefore, the person walking appears not to be important for these maps. In addition, the person is wearing gray and dark green clothes, which blend him in with the background. Instead, the local edge map emphasizes the sky which is bright and the leaves which have a lot of texture. The global texture map highlights the high contrast regions, that is, the silhouette of the black truck and the sky. It is only the disparity map (shown in Figure 3.8(c)) which recognizes that the individual is close to the cameras and, therefore, highlights him. When the three maps are combined as shown in Figure 3.8(e), the resulting bounding box manages to feature the person walking which is important in the frame.

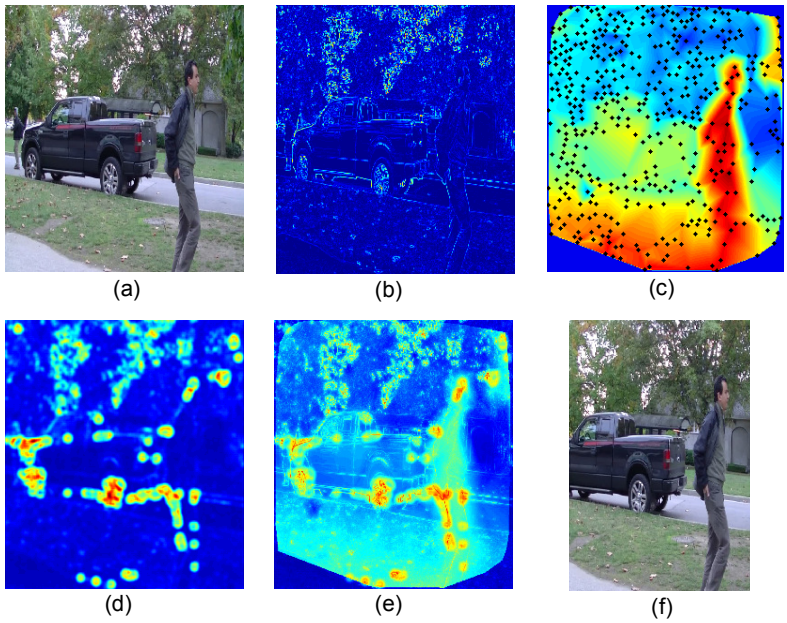


Figure 3.8. A frame from the "Black Truck" sequence. (a) Original frame (horizontally squeezed since it is the left half of the side-by-side 16:9 3D frame); (b) local edge saliency map s_l; (c) disparity saliency map s_d; (d) global texture saliency map s_g; (e) combined saliency map s_c based on all three maps; (f) resulting cropped frame with a 4:3 aspect ratio. In this case, the local edge map and the global texture map do not recognize the person as a salient object. The disparity map, however, identifies its proximity to the cameras and highlights it. The combined map includes all the salient points in the frame and the bounding box is chosen accordingly.

Some extreme targeting aspect ratios, such as cropping from 16:9 to 1:2, were also used to test the performance of our algorithm. For the case of 1:2 aspect ratio, *w* equals 35 pixels was used in the shrinking and expanding step. Results verified that the most salient objects were remained in the reframed video sequences. An example is provided in Figure 3.9, where the person is identified as the most prominent object and is centered in the cropped frame, even with the very narrow cropping window.

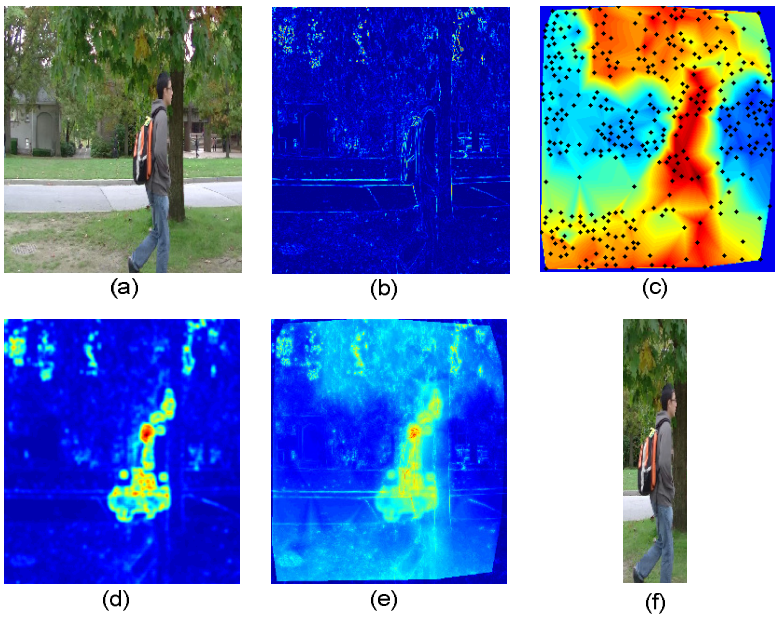


Figure 3.9. A reframing example using an extreme aspect ratio, demoed on a frame from the sequence "Run and Jump." (a) Original frame (horizontally squeezed since it is the left half of the side-by-side 16:9 3D frame); (b) local edge saliency map s_l; (c) disparity saliency map s_d; (d) global texture saliency map s_g; (e) combined saliency map s_c; (f) resulting cropped frame with a 1:2 aspect ratio.

3.3 Subjective Evaluations

In our subjective evaluations, we used fifteen representative stereoscopic video sequences that feature various scenarios, from capturing a single still object to tracking several people walking (towards the same direction and towards the opposite directions simultaneously); from having people running at a far distance to having the same person

running close to the camera. Some videos were recorded with a handheld camera and others had the camera mounted on a tripod. Most of the videos feature people working, playing, or walking, and we included both indoor and outdoor sequences. The lengths of the videos are from 10 to 43 seconds.

The features of the test video sequences are listed in Table 3.1. For each test sequence, the thumbnail of the left view of a representative frame is shown in Figure 3.10.

Table 3.1: Features of the 3D test video sequences

Index	Video Name	Camera Motion	Scene Complexity
1	Playground	Slightly shaky	Medium
2	Kids Playing	Zoom & shaky	High
3	Magic Rope	None	Low
4	Black Truck	Tracking	High
5	Main Mall	Pan & zoom	High
6	Run and Jump	None	Medium
7	Tree	None	Low
8	Greeting	None	High
9	McMillan Building	Pan & shaky	High
10	Lab	Slightly shaky	Low
11	UBC Flag	Pan & shaky	Low
12	Bicycle	Zoom & shaky	Low
13	Girls Walking	Tracking	High
14	Talk	Minor tracking	Medium
15	Lounge	Tacking	Medium

The subjective test was conducted on a 46” 3D LED monitor (Hyundai S465D). The display has a resolution of 1920×1080, and is paired with circular polarized glasses. The

viewing conditions for the subjective assessment were set up based on the recommendation in Section 2.1 of the ITU-R BT.500-11 [43].

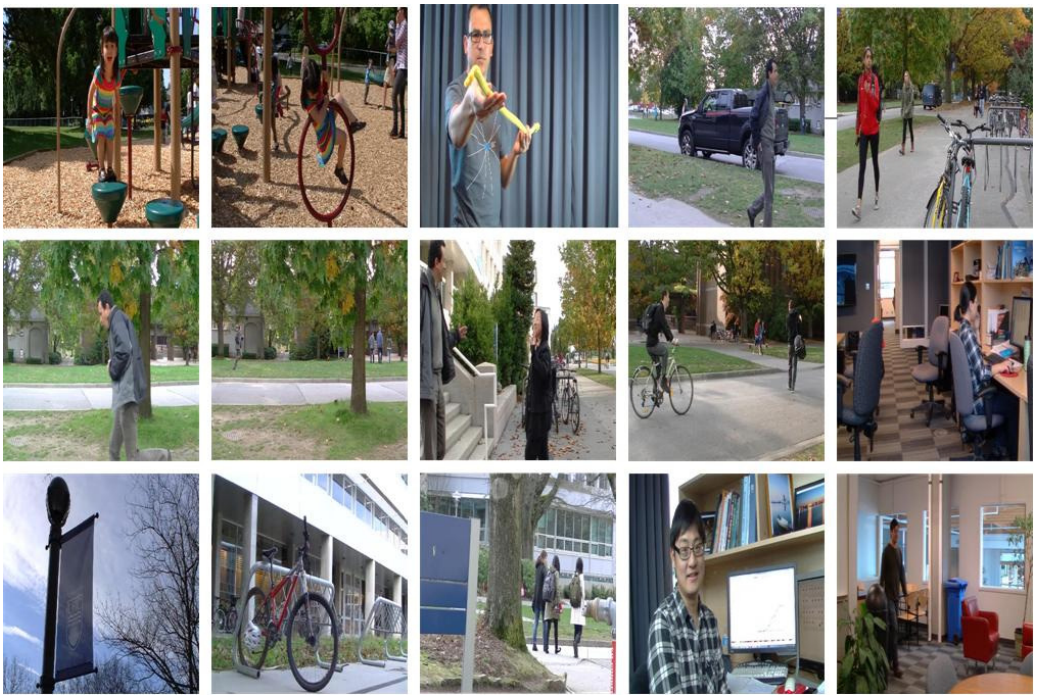


Figure 3.10. The left view of representative frames of our 3D test video sequences. In reading order: video 1 to video 15.

For comparison purposes, we provided two reframed versions for each video. One generated from our proposed scheme, and the other obtained by cropping each frame equally from the left and right sides, namely, the centered cropping scheme. Centered cropping is a popular and straightforward reframing technique. Since most of the important objects in a video are located near the centre of the frame when capturing, this scheme gives reasonably good results and it is a good reference point for comparing our approach.

For each video, we first showed its original version with a 16:9 aspect ratio, followed by two reframed videos of a 4:3 aspect ratio. The two reframed versions were shown in a

random order. Before each video was played, a four-second interval of a 2D mid-grey image with the index of the upcoming video was shown as a relaxation period. After each reframed video was shown, another four-second interval of a 2D mid-grey image was used as a grading period. We ran a training session at the beginning of the test using three videos that are different from the test sequences. The training session was arranged to familiarize the subjects with the test procedure as well as the quality range of our reframed videos, without imposing their quality.

The observers were asked to rate how well each reframed version preserves the important regions and minimizes viewing discomfort such as window violation in a stereoscopic 3D scene. The reframing quality is rated on the five-point discrete grading scale, where scales 5 to 1 respectively represents “excellent”, “good”, “fair”, “poor”, and “bad”.

Eighteen non-expert observers, including six females and twelve males, participated in the subjective test. Their ages ranged from 19 to 61, with an average age of 32.5. Three observers were identified as outliers based on the screening guidelines provided in Section 2.3.1 of annex 2 of ITU-R BT.500-11 recommendation [43]. Therefore, only scores of the fifteen valid subjects were used in the following analysis. We take the average score across all valid fifteen observers for each reframed video as the mean opinion score (MOS). Confidence intervals are used to indicate the reliability of an estimated mean opinion score. The Student's t-tests are used to compute confidence intervals with the significance level being 95%.

For the fifteen test video sequences, the results of the centered cropping algorithm and our proposed algorithm are shown in Figure 3.11. Our smart reframing technique outperforms the centered cropping scheme on all fifteen sequences. The performance gain ranges from 0.133 to 2.80 scales in MOS. An average gain of 1.14 scales out of 5 in MOS is achieved over all fifteen test videos.

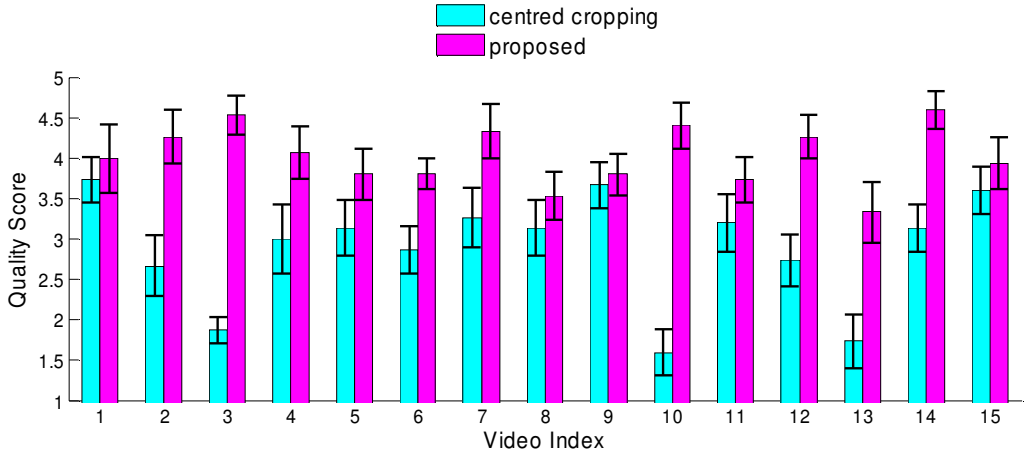


Figure 3.11. Comparison of the mean opinion scores and their confidence intervals of the fifteen test video sequences.

From Figure 3.11, we observed that sequences 3 and 10 received very low scores for the centered cropping algorithm and very high scores for the proposed algorithm. This is because the salient objects, such as the person's elbow in video 3 and the computer screen in video 10, are partially or totally cropped by the centered cropping technique, resulting very bad subjective quality for the corresponding video. The bad reframing is especially noticeable when it appears in a significant portion of the sequence. On the other hand, the videos generated by our technique are of much higher quality in comparison. This provides us an insight that a perfectly reframed video does not appear to be surprisingly good, whereas a video with salient regions cropped off is very noticeable and annoying to the viewers.

Figure 3.11 also shows that sequences 1, 8, 9, 11, and 15 have received similar ratings between the two algorithms, with our proposed method having slightly higher MOS. By carefully examining the videos, we found that most of the original test sequences tend to keep the objects of interests in the center of the frame and track them when they move. This applies to sequences 1, 9, 11, and 15. For these sequences, the centered cropping algorithm naturally keeps the prominent area and results in a good performance. Video 8, on a contrary, was captured on a tripod without any panning or zooming. It features two people entering the scene from opposite directions, greeting each other, and leaving the scene from opposite directions again. This is a tough sequence for any reframing technique, even for human observers. When two important people are located at the frame boundary, one on the left and the other on the right, there is no obvious choice for what person to follow during reframing. Our algorithm chose to follow the person that is closer to the camera. This receives acceptable subjective results. The centered cropping algorithm, on the other hand, keeps it as a steady scene. It offers reasonable results, since the two people quickly entered and left the scene even in the original video, leaving very limited additional information to track for. In other words, keeping a person half a second longer in the scene does not affect the overall quality.

Lastly, it is worth comparing videos 6 and 7. They were captured at the same location, both using a tripod, and mainly focusing on the same person. The main difference is that in video 6 the person ran and jumped close to the camera, while in video 7 he was far from the camera until the very last two seconds. Our algorithm believes that the person in video 6 is the salient object and keeps tracking him, while in video 7 it treats the waving tree as the most salient object before the person ran closer to the camera and drew more

attention. Subjective tests verified that these reframing results closely reflect the viewers' opinion.

3.4 Conclusions

In this chapter, we have proposed a novel and complete pipeline for smart 3D video reframing. This solution allows us to display high quality stereoscopic content on screens with different aspect ratios than the one chosen for the original content.

We first proposed a bottom-up 3D visual attention model that identifies the prominent regions in a stereoscopic 3D video frame. The model combines disparity, edges, motion, luminance, and chrominance information to generate a saliency map.

We then developed an automatic reframing approach to create a bounding box for each frame based on the saliency map of the current frame and also saliency maps associated with the neighbouring frames. Special attention was paid to avoid the important objects from being cropped or located right at the border of the new window. In addition, the temporal jerkiness of the cropping window was avoided. This was achieved by keeping the previous location of the bounding box instead of the one obtained with the current saliency map if the energy changes were below a certain threshold. In addition, a temporal low-pass filter was employed to the bounding box locations in order to further ensure the temporal smoothness of the cropping locations.

The proposed algorithm is easy to implement and computational efficient. Although our current Matlab implementation does not provide a real-time performance, the algorithm will be implemented so that it is suitable for real-time streaming. The results

show that our proposed scheme is very effective and robust for a great variety of stereoscopic video sequences. It works well for 3D videos that are captured with a tripod or handheld, still or panning, indoor or outdoor, with slow or fast motion, simple or complex scene. Subjective tests show that our algorithm outperforms the centered cropping algorithm by 1.14 out of 5 scales on average.

4 Color Correction for Saturated Pixels

The past two chapters have dealt with 3D capturing and display issues. In this chapter we move to an enhancement problem, namely correction of clipped pixels in LDR color images and videos.

As stated earlier in Chapter 1, 3D video systems can only be a lasting success if the perceived image quality and viewing comfort are significantly better than those of conventional 2D systems. A more realistic reproduction of contrast and color using HDR technologies is desirable. Hence, it is of great interest to combine the 3D immersive experience with HDR capabilities in order to produce a true to life viewing experience. Since the development of HDR capturing technology is still at the early stages generation of HDR content is a challenge. One solution that will help enable this market at these early stages is the creation of HDR content from conventional LDR images and videos. This is possible by using efficient inverse tone mapping (ITM) techniques. One challenge that affects the quality of the resulting HDR content in this process is color distortion that is due to saturation in the original LDR content and results in disturbing perceptual artifacts when shown on HDR displays. The problem is more severe for 3D HDR. Hence, designing algorithms that eliminate the LDR to HDR color distortion is of high importance.

In this chapter, we propose two novel and effective color-correction methods for LDR content. One, described in Section 4.1, is based on the ZB algorithm (introduced in Section 1.4.1) with additional emphasis on the image spatial correlation. The other, presented in Section 4.2, utilizes the strong correlation in chroma channels between saturated pixels and their surrounding unsaturated pixels.

4.1 A Fast Bayesian-Based Color Correction Method

As described in Section 1.4.1, the ZB algorithm uses all unsaturated pixels in an image to estimate the prior distribution. The inter-channel correlation is used, but not the spatial intra-channel correlation. In order to utilize the strong spatial correlation of images as well as the inter-channel correlation, we propose a modified desaturation algorithm, which uses local statistics for correcting each disconnected saturated region.

4.1.1 Proposed Color Correction Method

In the proposed algorithm, we first identify the clipped pixels and color channels using a simple threshold. A binary image A is generated to indicate the saturated and unsaturated pixels in an image:

$$A(z) = \begin{cases} 1, & \text{when the pixel at } z \text{ is saturated} \\ 0, & \text{when the pixel at } z \text{ is unsaturated} \end{cases} \tag{4.1}$$

Next, we find the set of pixels that are close to the saturated pixels for computing prior distribution model. By eliminating the pixels far from all saturated regions, the statistics of the selected pixel set should better resemble that of the saturated regions than using all unsaturated pixels in the image.

Since the saturated regions often have various sizes and irregular shapes, dilation is a good choice to find pixels in the neighborhood, regardless the sizes or shapes of the saturated regions. Dilation is defined in terms of set operations. The dilation [97] of A by B, denoted $A \oplus B$, is defined as:

$$C = A \oplus B = \{ z \mid (\hat{B})_z \bigcap A \neq \phi \} \tag{4.2}$$

where A is the binary image defined in (4.2), B is the structuring element that determines the scale and orientation that the dilation operation "grows" or "thickens" objects (i.e., saturated regions) in A. $(\hat{B})_z$ is the reflected and translated B, and ϕ is the empty set. As a result, C includes all saturated pixels and their surrounding unsaturated pixels.

In order to take advantage of the strong local spatial correlation, pixels for estimating prior distribution model need to be localized to reflect the statistics of the different saturated regions in different areas of an image. A simple separation of disconnected objects in C groups all saturated pixels and their surrounding pixels into several local regions, i.e., C_i, where i = 1, 2, 3, …. Each region is likely to have similar statistics. An example of dilation process is given in Figure 4.1. The baby image is shown in part (a).

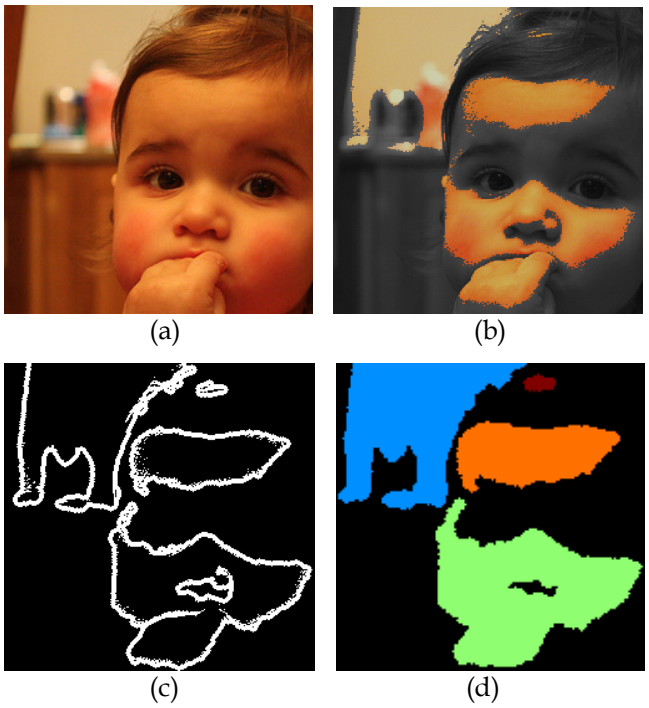


Figure 4.1: Generating surrounding regions by dilation using the XDN I algorithm. (a) A portion of the baby image, (b) the saturated areas, (c) surrounding pixels found by dilation, and (d) separated saturated areas and their local surrounding regions; each color represents a region for which a separate set of local statistics will be calculated.

Part (b) shows the saturated regions in color superimposed on the image luma. The surrounding areas of all of saturated regions are shown in white in part (c), and part (d) shows different saturated areas and their local surrounding regions by color.

Let S_i and U_i respectively denote the sets of saturated and unsaturated pixels in C_i. We calculate separate statistical parameters $\mu_s, \mu_k, V_s, V_k + V_{e_k}$, and V_{sk} for each region C_i, using only the unsaturated pixels U_i in its surrounding region. After the local parameters are calculated for each region, the rest of the correction is performed as in the ZB algorithm, only with the global statistics replaced by the local ones of the surrounding unsaturated pixels U_i. That is, the saturated channel(s) in S_i is corrected using equations (1.2), (1.3), and (1.4), where the statistical parameters are computed using pixels in U_i. By using these local pixels when computing the prior distribution, the spatial correlation is well integrated.

We experimentally chose the optimal structuring element *B* in (4.2) to be a disk with radius four. This structuring element results in an isotropic extension of the saturated regions, with a reasonable number of pixels in the surrounding region, which offers good local statistical information for computing the prior distribution of the RGB color channels for all of our test images.

Compared with the ZB original algorithm, the only additional steps necessary in our method are the dilation operation and splitting the dilated binary image into disconnected regions. Since these are only performed once, the extra cost is minimal. In our method, some computations are saved in calculating the statistical parameters in our method, as

they only need to be computed over the surrounding region pixels, compared to being calculated over the whole image in the ZB algorithm. Since the surrounding regions are usually a small subset of the entire image, our method has lower complexity for that step. There is a small amount of overhead for keeping tracking separate statistics of different surrounding regions. Overall, our method has very similar complexity to the ZB algorithm.

4.1.2 Experimental Results

In order to objectively test our method, we use conventional eight bits per channel color images. Thumbnails of our test images are shown in Figure 4.2.

Figure 4.2: Thumbnails of our test images. In reading order: girl, landscape, shoes, mountain, baby_girl, sunset, kodim03(caps), kodim05(motorcycles), kodim06(boat), kodim12(beach), kodim16(lake), kodim21(lighthouse), kodim23(parrots).

We introduce saturation into the images by clipping the *R*, *G* or *B* values that are above a threshold (e.g., 255×0.8). Then, we enhance the saturated images using the ZB algorithm, the only color correction algorithm for clipped pixels that we are aware of, and our proposed algorithm, which we shall refer to as the "XDN I" algorithm. We compare the corrected results to the original images without clipping.

To evaluate the algorithm performance in terms of image fidelity, we use three popular objective quality metrics that are mentioned in Section 1.1.4. More specifically, we compute the PSNR (averaged over *R*, *G*, and *B* channels), the CIELAB ΔE [101] (averaged over the saturated pixels), and the S-CIELAB [102] (averaged over the saturated pixels) of each test image for: 1) the saturated image, 2) the desaturated image generated by the ZB algorithm, and 3) the desaturated image produced by our improved XDN I algorithm. The quality comparison for a set of representative images is listed in Table 4.1. Note that S-CIELAB is a distance measure; a lower value means better quality. From the table, we observe that while both the ZB algorithm and our proposed XDN I algorithm enhance the saturated color channels, the XDN I method outperforms the ZB algorithm by an average of 2.61 dB in PSNR, 2.74 in CIELAB ΔE, and 0.46 in S-CIELAB over the test images. The XDN I algorithm performs better than the ZB algorithm, especially for images where the local statistics are inconsistent with the global statistical model.

Subjective quality of the desaturation algorithms is also evaluated and shown in Figure 4.3. For each representative image, we show (from left to right) the original

Table 4.1: Objective quality comparison between the ZB and XDN I algorithms

Image	PSNR (in dB)			CIELAB ΔE			S-CIELAB		
	Clipped	ZB	XDN I	Clipped	ZB	XDN I	Clipped	ZB	XDN I
girl	42.23	39.70	48.54	6.68	9.39	2.82	0.98	1.51	0.40
landscape	25.66	28.69	31.73	11.79	9.27	6.19	1.87	1.41	0.87
baby_girl	32.15	29.06	38.87	8.40	11.39	3.27	1.24	1.86	0.43
mountain	29.98	34.27	40.81	11.52	5.87	2.81	1.65	0.91	0.30
shoes	25.34	32.93	32.00	13.03	6.32	5.36	2.00	1.02	0.82
sunset	21.73	20.76	25.70	18.62	16.96	11.72	3.04	3.47	2.02
kodim03 (caps)	34.34	35.24	36.62	12.37	12.25	9.83	2.30	2.25	1.82
kodim05 (motorcycles)	33.62	35.74	36.50	13.46	11.00	10.79	2.25	1.83	1.77
kodim06 (boat)	25.22	28.22	26.12	17.45	10.26	14.46	3.44	1.97	2.89
kodim12 (beach)	28.41	33.65	31.76	11.04	4.55	3.99	2.08	0.81	0.74
kodim16 (lake)	35.07	35.85	36.51	11.36	10.42	8.36	2.16	1.98	1.54
kodim21 (lighthouse)	32.40	33.56	34.24	16.43	13.35	11.72	3.03	2.51	2.22
kodim23 (parrots)	29.63	31.16	33.42	10.38	8.54	6.15	1.70	1.40	1.01
Average	**30.44**	**32.22**	**34.83**	**12.50**	**9.97**	**7.50**	**2.14**	**1.76**	**1.30**

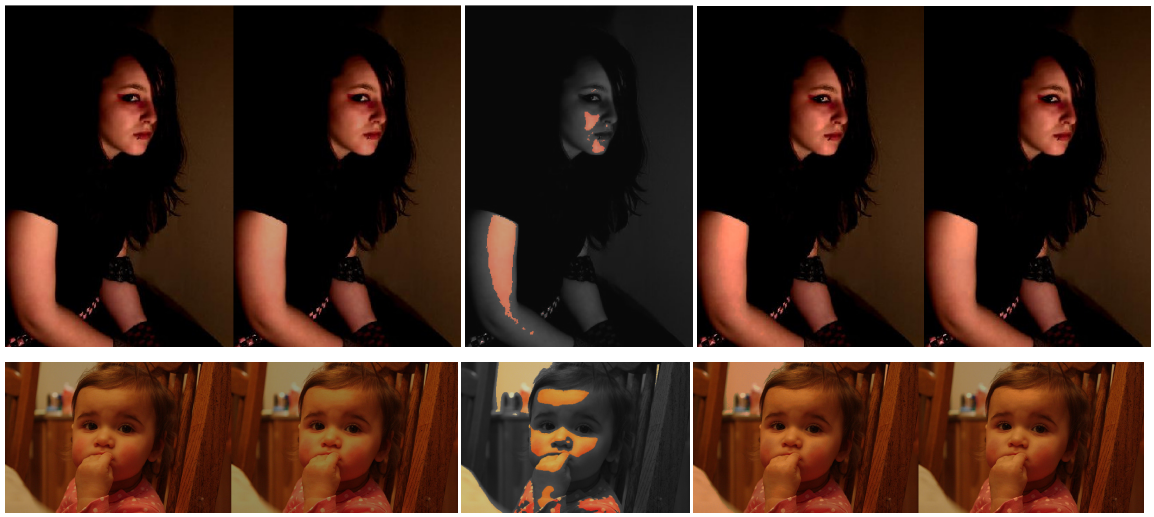


Figure 4.3: Results of clipped pixel enhancement for images girl and baby using the XDN I algorithm. For each row, we show (from left to right) the original image, saturated image, saturated areas superimposed on the image luma, desaturated image using the ZB algorithm, and desaturated image using the XDN I algorithm.

image, saturated image, saturated areas superimposed on the image luma, desaturated image using the ZB algorithm, and desaturated image using our XDN I algorithm. From Figure 4.3 we observe that the saturated images have color distortions due to clipping. While the ZB algorithm corrects such distortion for most saturated regions, it miscorrects some areas and results in further color distortion. An over-correction example can be seen in the background area of the baby image. Such miscorrection happens when the statistical model of a saturated region differs from the global statistical model of the unsaturated pixels in the image. Compared to the ZB algorithm, our XDN I algorithm gives better or comparable subjective quality, while avoiding the blocky artifacts, such as the arm area in the girl image. The saturated pixels in the arm area have very different statistics from the dark portion of the image. Therefore, using the global prior distribution model derived from all the unsaturated pixels in the image results in poor performance. The XDN I algorithm uses nearby pixels to generate local statistics for the clipped pixels in the arm and leads to better results.

4.1.3 Conclusions

In this section, we have investigated correcting saturation in color-images. We have proposed a fast and improved Bayesian algorithm based on the ZB algorithm. Our XDN I method utilizes the images' strong spatial correlation in addition to the correlations between *R*, *G*, and *B* color channels. We use a dilation operation to find a surrounding area for each clipped region in the image, and use statistics calculated based on this surrounding region for correcting the saturated pixels. Experimental results show that the proposed XDN I method effectively corrects the saturated color images, and outperforms

the ZB algorithm in both objective and subjective image qualities. The quality gain is by an average of 2.61 dB in PSNR, 2.74 in CIELAB ΔE, and 0.46 in S-CIELAB.

4.2 An Effective Color Correction Method

As mentioned in Section 4.1, we proposed a modified Bayesian algorithm, namely XDN I, based on the ZB algorithm. Both the ZB and XDN I algorithms use the correlations between *R*, *G*, and *B* color channels, which may not be the most suitable way for exploiting the relationships among color pixels. The pixels could be more strongly correlated in the spatial domain and in some other color spaces. In this section, we propose another effective clipped-pixel enhancing algorithm, which we refer as the "XDN II" algorithm. It automatically restores both the luma and chroma of the clipped pixels. In the XDN II algorithm, we exploit the strong correlation in chroma between saturated pixels and their surrounding unsaturated pixels. Experimental results show that the XDN II algorithm outperforms the ZB and XDN I algorithms in both objective and subjective quality evaluations.

The rest of this section is structured as follows. Section 4.2.1 describes our proposed method for still images. Extensions of the algorithm to videos and 3D content are discussed in Sections 4.2.2 and 4.2.3, respectively. The experimental results are presented in Section 4.2.4. In Section 4.2.5, we conclude the proposed method and point out its potential applications.

4.2.1 Proposed Color Correction Method for 2D Still Images

In this section, we aim at restoring the lost information in over-exposed color images based on the strong spatial correlation in the chroma channels. The YCbCr color space is

designed so that the chroma channels will be smooth in local regions for most images. It has been shown that utilizing the smoothness property of chroma [88], [89], [90] is better than assuming luma is smooth [81], [86]. Figure 4.4 shows the normalized autocorrelation of *R*, *G*, *B*, *Y*, *Cb*, and *Cr* at lags of 0 to 25 pixels. Each point in the figure is an average value over the 24 true-color Kodak images [91]. From the graph, we can see that there is stronger auto-correlation for the *Cb* and *Cr* channels than the *R*, *G*, *B*, and *Y* channels. Exploiting the strong spatial correlations in the *Cb* and *Cr* channels has more potential than exploiting the correlations in the *R*, *G*, *B*, or *Y* channels. For this reason, in our XDN II algorithm we apply a chroma interpolation for the clipped pixels rather than directly correcting the *R*, *G*, and *B* signals.

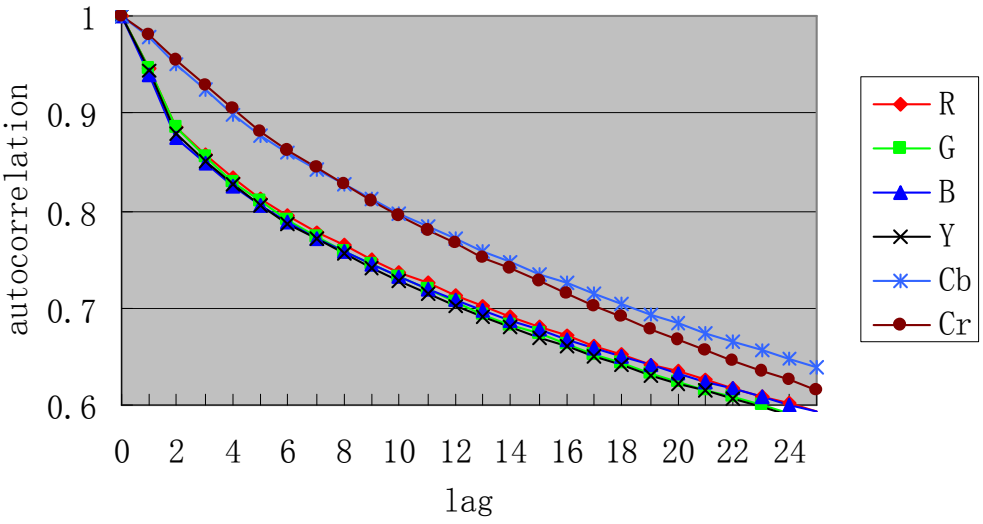


Figure 4.4: Normalized autocorrelation of *R*, *G*, *B*, *Y*, *Cb*, and *Cr* signals (average over 24 true-color Kodak images).

Our proposed XDN II method can be broken down into several steps, which are shown in the flowchart in Figure 4.5. First, we identify the clipped areas. Then we

partition each clipped area into smaller regions according to the chroma. We correct the chroma for each region, and then correct the corresponding RGB values for all clipped pixels. Afterwards we apply a smoothing process to the corrected RGB values. The last step involves "Enhancing the Contrast" and can be performed by using any existing inverse tone mapping process [80], [81], [92], which is not the focus of our work. A detailed description of these steps is given in the following subsections.

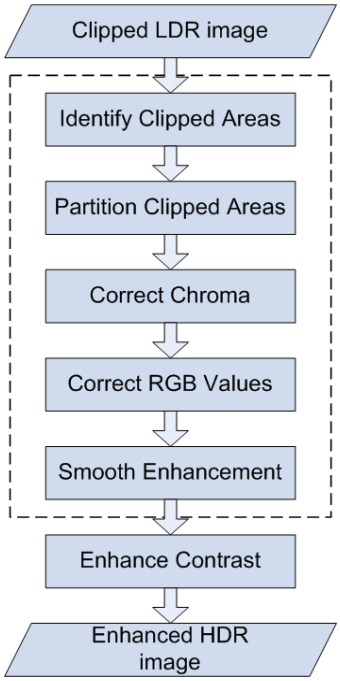


Figure 4.5: Flowchart of the XDN II algorithm.

4.2.1.1 Identify Clipped Areas

Before doing any enhancement, we first need to identify the clipped areas. One way of doing this is to simply select pixels from all three color channels that have the maximum value (e.g., 255 for 8-bit per channel images). Figure 4.6(a) shows a clipped image with a rectangular region of interest. Figure 4.6(b) shows the clipped area (within the region of interest) identified with a simple threshold (the maximum value, e.g., 255

for 8-bit per channel images). The white pixels represent the clipped area. As it can be seen, this simple approach often generates very small isolated clipped areas, and large clipped areas with small holes. The effect is due to image noise. The captured pixel values are determined not only by the light from the scene, but also by the camera response, sensor noise, and color filter array interpolation. The in-camera processing adds noise to a pixel value and, consequently, a clipped pixel may have a value slightly lower than the maximum value. For this reason, we first apply a bilateral filter [93], [94], [95] to remove the noise. Then, a threshold τ is applied to each color channel of a pixel to identify clipped pixels and channels. We experimentally selected τ to be 252.5 for 8-bit per channel images herein. Figure 4.6(c) shows the clipped area in the region of interest identified by the XDN II algorithm. We observe that the clipped area in Figure 4.6(c) is quite clean and more appropriate for subsequent color correction compared to that in Figure 4.6(b). Note that the bilateral filter is used only for identifying the clipped areas. The original un-filtered image is used in all of the following steps to avoid losing detail from the image.

Figure 4.6: Example of clipped areas in the XDN II algorithm. (a) Clipped image with a rectangular region of interest, (b) clipped areas in the region of interest identified with a simple threshold (*R*, *G*, or *B* = 255), and (c) clipped areas in the region of interest identified with the XDN II algorithm.

4.2.1.2 Partition Clipped Areas

The purpose of partitioning the clipped areas is to group the clipped pixels into regions with similar chroma before correcting the color for each region. We first partition the clipped areas into spatially disconnected regions, which probably belong to different surfaces and have different chroma. Each region may still contain multiple clipped objects that have different colors. We segment each region according to its chroma. To eliminate the illumination differences on the same color surface, we consider only chroma quantities, i.e., *Cb* and *Cr*, in the segmentation. For simplicity, in order to segment the region we choose to use either the *Cb* or *Cr*, whichever has a larger variance within the considered region.

The pixels with heavier saturation (i.e., 2- or 3-channel saturated pixels) are considered as one sub-region, where the color is potentially heavily distorted. The 1-channel saturated pixels are further segmented using a histogram-based multi-threshold algorithm presented in [97]. This segmentation algorithm often results in a few large and many small sub-regions. Finally, we merge these small regions or the regions without a valid surrounding region (Note: surrounding region will be explained shortly in Section 4.2.1.3) with their neighboring clipped regions. If more than one neighboring clipped region exists, the current region is merged with the neighboring region that is the closest to the current region in chroma.

An example of clipped area partition is given in Figure 4.7, where connected clipped areas (i.e., white pixels in Figure 4.7(a)) are partitioned into smaller regions (shown and

numbered in Figure 4.7(b)). Each region has similar chroma. This partitioning is essential for the subsequent color correction steps.

(a) (b)

Figure 4.7: Example of clipped-area partition in the XDN II algorithm. (a) The clipped areas before partition (white pixels), and (b) the clipped areas partitioned into regions with similar chroma.

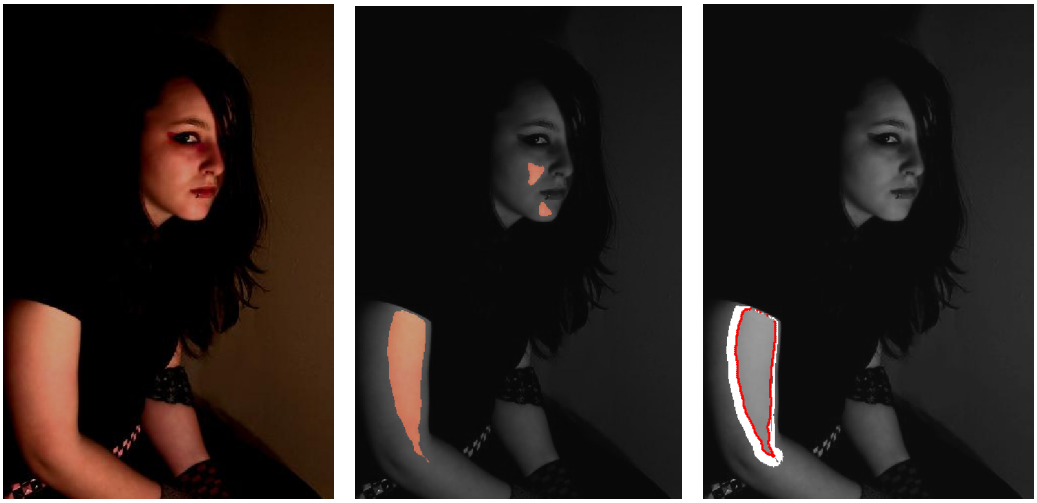
4.2.1.3 Correct Chroma

As explained earlier, the *R*, *G*, and *B* values are strongly affected by the illumination of the area. There are much stronger spatial variations in *R*, *G*, and *B* values than in chroma. Although the *RGB* and chroma in clipped areas can both be estimated using an interpolation method given their neighboring unclipped areas, a smooth signal, like the chroma, may be more accurately estimated, since there is less spatial variation associated with such a signal. For this reason, we chose to estimate the chroma values in a clipped region by smoothly interpolating the chroma of neighboring unclipped pixels. Once the chroma values are estimated, then we use them to calculate the corrected *R*, *G*, or *B* values in the clipped regions.

In order to correct a clipped region, we first attempt to find an unclipped region with similar chroma next to it. We select neighboring pixels with similar color to the clipped

region as seed points. This is done by first choosing the unclipped or already corrected neighboring pixels with gradients of both chroma channels less than a threshold (experimentally, we determined 2.5 works well). Then, starting from each seed point, we apply a region growing algorithm shown in [97] to both *Cb* and *Cr*, and the intersection of the two obtained regions is a surrounding region with similar chroma to the clipped region. Since there may be small chroma variations within each clipped region, we take the union of all surrounding areas obtained from different seed points as the surrounding region for a clipped region. If the resulting surrounding region consists of only very few pixels, then the clipped region is considered as a light source or a specularly reflected area. In this case, we enhance only the luma signal.

Figure 4.8 shows an example of a surrounding region associated with the clipped region on the girl's arm. We observe that most unclipped pixels in the arm area (with similar chroma as the clipped region) that are close to the clipped region are selected as



(a) (b) (c)

Figure 4.8: Example of a surrounding region in the XDN II algorithm. (a) Clipped image, (b) clipped areas (in color) superimposed on the image luma, and (c) the surrounding region (white pixels) for the clipped area on the girl's arm.

the surrounding region, which is used for the chroma estimation of the clipped pixels on the arm (Figure 4.8(c)).

The *Cb* and *Cr* values of the clipped region could be interpolated from its surrounding region. A problem arises from the fact that the surrounding region is irregularly shaped with some "missing" pixels which cannot be used in the interpolation because they are either clipped pixels, or non-clipped pixels with different chroma to the current clipped region. A common interpolation approach is to use convolution (filtering). However, traditional convolution does not work when there are missing samples within the convolution mask.

For the reason stated above, we use normalized convolution [98] instead, which allows for missing samples by adjusting the filter weights to use only the valid samples that fall within the convolution mask. The idea of normalized convolution is to associate with each pixel a certainty component m expressing the level of confidence in the pixel measurement. The certainty map m has the same dimension as the image.

To make the discussion more pertinent to our problem at hand, that is interpolating chroma for saturated pixels, the normalized convolution can be expressed as follows:

$$\tilde{c}(x, y) = \frac{\left[c(x, y) \cdot m(x, y)\right] * h(x, y)}{m(x, y) * h(x, y)}, \tag{4.3}$$

where the certainty map $m(x,y)$ is 1 for the known pixels that are used in the interpolation, and $m(x,y)$ is 0 for the missing samples. The $c(x,y)$ and $\tilde{c}(x, y)$ represent the chroma channel signal (*Cb* or *Cr*) before and after the convolution, and $h(x, y)$ denotes the filter

for performing the convolution. Here, a Gaussian filter with a standard deviation 5 is used as the function $h(x, y)$.

The normalized convolution mask $h(x, y)$ has a finite size. Consequently, a pixel located near the center of the clipped region may not have any pixel in the surrounding region lying in its mask. Hence, the pixel value cannot directly be corrected. In order to solve this problem, we choose to use the already corrected clipped pixels together with the surrounding region as the known data for estimating the un-corrected clipped pixels. In other words, the certainty map used is:

$$m(x, y) = \begin{cases} 1, & \text{for unsaturated pixels in the surrounding region} \\ & \quad \text{and saturated pixels that have been corrected,} \\ 0, & \text{otherwise,} \end{cases} .$$

This helps to improve the smoothness of the corrected chroma in the clipped region. Since already corrected pixels are used in the normalized convolution, the pixel order within a clipped region is very important. Because estimation error could propagate, the pixels with potentially less error should be corrected first.

In order to describe the smoothing process, we define a few notations here. Let Ω denote the saturated pixel set, that is,

$$\Omega = \{(x, y) : R(x, y) \geq \tau, \text{ or } G(x, y) \geq \tau, \text{ or } B(x, y) \geq \tau\} .$$

Furthermore, we use Ω_1, Ω_2, and Ω_3 to respectively represent the sets of clipped pixels with 1, 2, and 3 saturated channels.

Since the 1-channel saturated pixels Ω_1 tend to have less color distortion, and, hence, less estimation error than 2- and 3-channel saturated pixels Ω_2 and Ω_3, we first correct the pixels in Ω_1, followed by pixels in Ω_2, and lastly pixels in Ω_3. Since clipped pixels that are close to the surrounding region tend to have a strong correlation with the surrounding unclipped pixels, there is small estimation uncertainty, i.e., a small degree of error estimation for such pixels. For this reason, within each saturation category, we also sort the clipped pixels according to their distances to the nearest surrounding pixels, and first correct the ones closer to the surrounding region.

Figure 4.9 shows the results of chroma correction using our method described in this sub-section. Since the clipping in this image happens mostly in the red channel, we show the correction results by presenting the *Cr* channel (before and after correction) in Figure 4.9. We can see in the circled area that the clipped image (b) is darker than the un-clipped image (a), resulting in blocky distortion of the *Cr* channel. While the corrected chroma *Cr*, shown in part (c), is very close to that of the un-clipped image. The above chroma-correction result can be observed more easily in parts (d) and (e), where the difference between (a) and (b), and the difference between (a) and (c) are shown.

Figure 4.9: Example of chroma correction in the XDN II algorithm. The *Cr* channel of (a) un-clipped image, (b) clipped image, and (c) corrected image using the XDN II algorithm. The difference between (a) and (b) is shown in (d), and the difference between (a) and (c) is shown in (e).

4.2.1.4 Correct RGB Values

We calculate the missing *R*, *G*, or *B* values in each clipped region based on the estimated *Cb* and *Cr* values (calculated in the previous step) and the unsaturated *R*, *G*, or *B* values in that region. We elaborate this correction process for the following three different scenarios, i.e., Ω_1, Ω_2, and Ω_3.

4.2.1.4.1 Correct 2-channel saturated pixels

Correction of 2-channel saturated pixels is the most straightforward scenario. We know the conversion from RGB to YCbCr, introduced in the ITU-R BT.601 [99], is:

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 0.2568 & 0.5041 & 0.0979 \\ -0.1482 & -0.2910 & 0.4392 \\ 0.4392 & -0.3678 & -0.0714 \end{bmatrix} \times \begin{bmatrix} R \\ G \\ B \end{bmatrix} + \begin{bmatrix} 0.0627 \\ 0.5020 \\ 0.5020 \end{bmatrix}. \tag{4.4}$$

The above RGB and YCbCr values are within a range of 0.0 to 1.0. From (4.4), we have

$$Cb = [-0.1482 \quad -0.2910 \quad 0.4392] \times [R \quad G \quad B]^T + 0.5020, \tag{4.5}$$

$$Cr = [0.4392 \quad -0.3678 \quad -0.0714] \times [R \quad G \quad B]^T + 0.5020. \tag{4.6}$$

When two channels are clipped, then one of the *R*, *G*, and *B* values, say *U* (which stands for the unsaturated channel), is known and the two clipped channels, say S_1 and S_2 (which stands for the saturated channels), are unknown and need to be solved for. The *U*, S_1, and S_2 are all components in the three color channels [*R*, *G*, *B*]. The corrected values of the two saturated channels can be uniquely solved for using the two equations (4.5) and (4.6). Therefore, we have:

$$\tilde{S}_1 = f_1(U, Cb, Cr),$$
$$\tilde{S}_2 = f_2(U, Cb, Cr),$$

where f_1 and f_2 are functions of U, Cb, and Cr, and $\tilde{S}$ denotes the corrected value of color channel S. Note that we do not use Y to correct the RGB color channels, since Y is distorted when any color channel is clipped. The functions f_1 and f_2 can be derived uniquely from the RGB to CbCr conversion equations (4.5) and (4.6).

As an example, let us consider a case where R and G are the two clipped unknown channels in a saturated pixel, and B is the unclipped channel. From (4.5) and (4.6), we can solve for $\tilde{R}$ and $\tilde{G}$ given Cb, Cr, and B as follows:

$$\begin{aligned}\tilde{R} &= \frac{(Cb - 0.5020 - 0.4392B)\times(-0.3678) - (Cr - 0.5020 + 0.0714B)\times(-0.2910)}{(-0.1482)\times(-0.3678) - 0.4392\times(-0.2910)} \\ \tilde{G} &= \frac{(Cb - 0.5020 - 0.4392B)\times 0.4392 - (Cr - 0.5020 + 0.0714B)\times(-0.1482)}{(-0.2910)\times 0.4392 - (-0.3678)\times(-0.1482)}\end{aligned}.$$

Any other two channel saturated pixels can be corrected in the same fashion.

4.2.1.4.2 Correct 1-channel saturated pixels

Correction of 1-channel saturated pixels is similar to correcting 2-channel saturated pixels. Since there is only one unknown value S, and two equations, (4.5) and (4.6), the value can be estimated twice by using the corrected Cb and Cr, respectively, as well as the two unsaturated channel values U_1 and U_2,. Then, we simply take the average of the two estimations as the corrected value of the saturated channel. The estimation process can be described as:

$$\tilde{S} = \frac{S_1 + S_2}{2}, \qquad \text{where}$$
$$S_1 = f_3(U_1, U_2, Cb),$$
$$S_2 = f_4(U_1, U_2, Cr),$$

where the functions f_3 and f_4 are derived from (4.5) and (4.6), respectively.

As an example, let us consider a case where R is the clipped unknown channel, and G and B are the unclipped known channels. The corrected value $\tilde{R}$ is computed as follows:

$$\tilde{R} = \frac{R_1 + R_2}{2}, \qquad \text{where}$$
$$R_1 = \frac{Cb - (-0.2910) \times G - 0.4392 \times B - 0.5020}{-0.1482},$$
$$R_2 = \frac{Cr - (-0.3678) \times G - (-0.0714) \times B - 0.5020}{0.4392}.$$

Any other channel saturated pixels (i.e. if G or B is clipped) can be corrected in the same fashion.

4.2.1.4.3 Correct 3-channel saturated pixels

In the case of 3-channel saturated pixels, there are three unknown variables. Hence, three equations are needed to solve the corrected R, G, and B values. We first estimate the luma Y value of the 3-channel saturated pixels based on the surrounding region. We fit the clipped region and its surrounding area with a 2D Gaussian function. Unlike many other surface-fitting methods (e.g., [86]), we do not enforce any assumptions on the location or rotation of the 2D Gaussian function. By not assuming the center of the Gaussian function as the centroid of the clipped region, we are able to handle more general and sophisticated clipping cases. For example, our model works well for the

situation where the brightest spot is not located near the center of the clipped region and the surrounding region only partially encloses the clipped area. In general, a 2D Gaussian function is of the form:

$$g(x, y) = Ae^{-[a(x-x_0)^2+2b(x-x_0)(y-y_0)+c(y-y_0)^2]} + B,$$

where A, B, a, b, c, x_0, and y_0 are the parameters, and $\begin{bmatrix} a & b \\ b & c \end{bmatrix}$ is positive definite.

The least squares surface-fitting problem can be solved using the following optimization form:

$$\underset{A,B,a,b,c,x_0,y_0}{\operatorname{argmin}} \quad \sum_{i=1}^{n} [Y_i - g(x_i, y_i)]^2$$

$$\text{Subject to: } \begin{bmatrix} a & b \\ b & c \end{bmatrix} \succ 0,$$

where (x_i, y_i, Y_i) is the ith pixel in the surrounding area, x_i and y_i represent the pixel location, and Y_i is the luma at pixel (x_i, y_i), and the symbol '$\succ$' stands for positive definite. In order to remove the constraint from the above optimization problem, we apply variable substitutions. A symmetric and positive definite matrix M can be decomposed into $M=LL^T$, where L is a lower triangular matrix [100]. For the matrix $\begin{bmatrix} a & b \\ b & c \end{bmatrix}$ in the constraint, we have $\begin{bmatrix} a & b \\ b & c \end{bmatrix} = \begin{bmatrix} l_{11} & 0 \\ l_{21} & l_{22} \end{bmatrix} \times \begin{bmatrix} l_{11} & l_{21} \\ 0 & l_{22} \end{bmatrix}$. Substituting a, b, and c using the new variables l_{11}, l_{21}, and l_{22}, the constraint is implied in the relation. Therefore, the optimization problem becomes unconstrained, as follows:

$$\underset{A,B,l_{11},l_{21},l_{22},x_0,y_0}{\operatorname{argmin}} \sum_{i=1}^{n}[Y_i - g(x_i, y_i)]^2 ,$$

where $g(x,y) = Ae^{-[\frac{l_{11}^2}{2}(x-x_0)^2 + l_{11}l_{21}(x-x_0)(y-y_0) + \frac{l_{21}^2+l_{22}^2}{2}(y-y_0)^2]} + B$.

The optimization can be solved with a standard least squares fitting algorithm. Once the parameters are estimated, the luma Y at the 3-channel saturated pixels can be computed by evaluating the Gaussian function. In the end, the corrected RGB values $\tilde{R}$, $\tilde{G}$ and $\tilde{B}$ are solved using (2) as follows:

$$\begin{bmatrix} \tilde{R} \\ \tilde{G} \\ \tilde{B} \end{bmatrix} = \begin{bmatrix} 1.16438356 & 0.00000030 & 1.59602688 \\ 1.16438356 & -0.39176253 & -0.81296829 \\ 1.16438356 & 2.01723263 & 0.00000305 \end{bmatrix} \times \left(\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} - \begin{bmatrix} 0.0627 \\ 0.5020 \\ 0.5020 \end{bmatrix} \right).$$

In the process of correcting saturated RGB values, we need to eliminate unrealistic corrected pixel values and obtain a stable enhancement algorithm. Hence, we set a lower bound α and an upper bound β as the multiplicative enhancement factor (i.e., the ratio between corrected value and clipped value) for each clipped pixel. We know the fact that the true values in the clipped channels should be greater than the clipped value. Therefore, the lower bound α is set to be greater than (or equal to) one. To ensure a smooth transition between the unsaturated and saturated regions, the lower bound α is acted as a smooth enhancement-factor mask to the saturated region. The mask can be denoted as:

$$\alpha = \begin{cases} (\alpha_0 - 1) \times \dfrac{d}{d_0} + 1, & \text{when } 0 < d < d_0 \\ \alpha_0, & \text{when } d \geq d_0 \end{cases},$$

where the enhancement factor α_0 is a constant and $\alpha_0>1$, the constant d_0 is the transition width, and d denotes the distance between the current saturated pixel and the closest unsaturated pixel in the surrounding region. The mask keeps the lower enhancement factor as α_0 for the pixels far from the unsaturated region, and gradually reduces the lower bound to 1 as the pixel gets closer to the unsaturated region.

4.2.1.5 Smooth Enhancement

The main purpose of color enhancement is to obtain visually plausible images and videos. Often, there are small jumps of enhanced values between adjacent 1-, 2-, and 3-channel saturated regions. This is because different strategies are used for Ω_1, Ω_2, and Ω_3 when calculating saturated RGB channels from the corrected *Cb* and *Cr*, as described in Section 4.2.1.4. As a result, a smoothing process near the region boundaries of Ω_1, Ω_2, and Ω_3 is needed to reduce disturbing contours and obtain natural looking enhanced images.

In order to smooth the boundary between regions Ω_i and Ω_j, where $i > j$, among the pixels near the region boundary, we choose to adjust the pixels in Ω_i, where the pixels have more saturated channels and relatively higher estimation errors than those in Ω_j). Figure 4.10 illustrates the smoothing process. We create a transition band with a width w_0 on the more saturated side (i.e., Ω_i) of the region boundary. The smoothed value at pixel (x_0,y_0) in the transition band is a linear combination of the estimated values (from the previous correction steps) at this pixel and its nearby region A_{x_0,y_0}. We define the area A associated with pixel (x_0,y_0) by first finding the pixel (x_1,y_1) that is closest to (x_0,y_0) and

also in the less saturated region Ω_j. The area A_{x_0,y_0} is composed of (x_1,y_1) and its surrounding pixels in Ω_i that are within a distance of 3 pixels from (x_1,y_1).

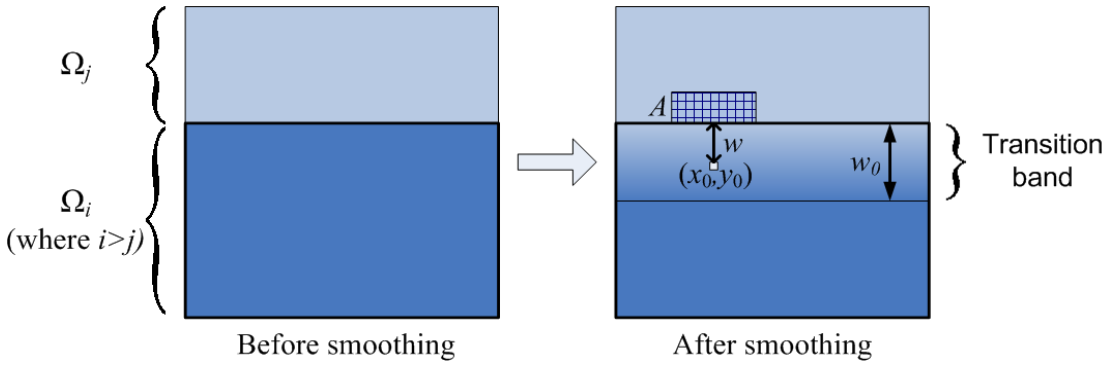


Figure 4.10: Illustration of the smoothing process between 1-, 2-, and 3-channel saturated regions, i.e., Ω_1, Ω_2, and Ω_3.

The adjusted saturated-channel value $\tilde{P}(x_0,y_0)$ at (x_0,y_0) is given by:

$$\tilde{P}(x_0, y_0) = P(x_0, y_0)\times\frac{w}{w_0}+\left(1-\frac{w}{w_0}\right)\times\frac{1}{N}\times\sum_{(x,y)\in A_{x_0,y_0}} P(x, y), \qquad (4.7)$$

where w_0 is the width of the transition band, w is the distance between (x_0,y_0) and (x_1,y_1), and N is the number of pixels in the area A_{x_0,y_0}. The parameter w_0 can be chosen to adjust the amount of the smoothing. A reasonable range of w_0 is 3 to 10 pixels. The adjusted pixel value $\tilde{P}(x_0,y_0)$ is a linear combination of pixel values at (x_0,y_0) and A_{x_0,y_0}. The reason we take a small area A_{x_0,y_0} rather than a single pixel in Ω_j is to make the transition band smoother, and avoid streaks in the band due to texture in Ω_j).

The effect of the smoothing process is illustrated in Figure 4.11. Part (a) shows the saturation-category map, with light gray, dark gray, and white representing Ω_1, Ω_2, and

Ω_3, respectively, and black being the unsaturated region. We observe that the enhanced image before the smoothing process has blocky artifacts between different saturated regions, while the enhanced image after smoothing appears more natural and visually pleasant.

(a) (b) (c)

Figure 4.11: Example of the smoothing effect in the XDN II algorithm. (a) The saturation-category map, (b) enhanced image before smoothing, and (c) enhanced image after smoothing.

4.2.2 Extension to Video Sequences

The previously proposed algorithm in Section 4.2.1 corrects saturated pixels in color images. When it comes to correcting video sequences, we may apply the algorithm to each video frame. With this approach, however, very annoying flickering problems will likely occur, due to the temporal inconsistency between consecutive frames. In this section, we describe an extension of the proposed algorithm to video sequences and ensure the temporal consistency.

We first identify and partition the saturated areas in every video frame, following the same procedures described in Sections 4.2.1.1 and 4.2.1.2. Then, we track the saturated regions over time. To do this, we match each segmented region, using its surrounding

unsaturated or corrected pixels, with pixels in the previous consecutive frame, based on the normalized cross correlation measurement. To save computations, the same surrounding region computed in Section 4.2.1.3 is used in calculating the normalized cross correlation. In the previous frame, a restricted search range of (-15, 15) pixels is used to reduce the computational cost. The location with the highest correlation value is considered to be the corresponding region in that frame. Performance evaluations have shown that if the highest correlation value is less than 0.95, it is safe to conclude that the same region does not exist in the previous frame, and consider this region appearing for the first time in the current frame. We record this link and the related motion vector, and create a linked region list over time for each saturated region.

Once the linked regions are identified, we apply two changes to ensure the temporal consistency of the corrected colors for video sequences. The first change ensures the spatial and temporal smoothness of the corrected chroma channels. Instead of using the 2D filter in (4.3) for still images, we apply a 3D Gaussian filter when correcting chroma. Hence, equation (4.3) is changed to:

$$\tilde{c}(x, y, n) = \frac{\left[c(x, y, n) \cdot m(x, y, n)\right] * h(x, y, n)}{m(x, y, n) * h(x, y, n)}, \tag{4.8}$$

where n is a frame index and all signals are extended to include a time dimension. The location coordinates x and y are the adjusted values based on the linked regions. The certainty map $m(x,y,n)$ is 1 for the unsaturated or corrected pixels in the linked regions, and 0 otherwise. We use a 3D Gaussian filter with a standard deviation 5 as the function $h(x, y, n)$.

To further prevent flickering caused by any of the subsequent processes described in Sections 4.2.1.4 and 4.2.1.5, we apply temporal filtering to ensure that the corrected RGB values are consistent between frames and changes very slowly over time. This is done in the very last step after applying the smooth enhancement to each frame as shown in Section 4.2.1.5. We temporally filter the values of the saturated channels by using a moving average filter of length 20 to the pixels at the same linked location. Performance evaluations have shown that this effectively removes the visible flickering artifact in corrected video sequences.

4.2.3 Extension to 3D Content

Simply applying the proposed algorithm to the left view and right view of a stereoscopic 3D video may result in color inconsistency between the two views. Although small degrees of color inconsistency are not noticeable by the human visual system [104], adjustment is needed to avoid large differences in color between the two views. Many color correction methods have been developed for stereoscopic and multiview video sequences [105]-[111]. They, however, mostly target videos where the colors of the entire view do not match those of the other view due to reasons such as imperfect calibration or variations in camera parameters. To eliminate the color inconsistency in small saturated regions, we choose to adjust the algorithm proposed in [110] for our situation.

For each saturated region, we need to match points between the left and right views. Then, the pixel colors will be fixed for each view as described in detail in [110]. To find reliable matching points, however, we note that the pixel values in the saturated regions

are inaccurate due to the color saturation and post-processing. Therefore, we use the neighboring unsaturated area to indirectly determine the matching points in the saturated regions. To do this, we first define a neighboring area, in which pixels are unsaturated and close to the saturated region both spatially and in chroma. Then, we match points between the left and right views in the neighboring area, and compute the parallaxes of the matching points. Next, we take the average of these parallaxes and consider it as the parallax of the saturated region. Based on this average parallax, we find the matching points in the saturated regions between the two views, and compute the average color over all matching points. Finally, the pixel colors are fixed based on a least-squares regression performed for each view in order to find a function that make the view most closely match the average color.

4.2.4 Experimental Results

In this section, we present some experimental results to show the effectiveness of the proposed XDN II algorithm for enhancing the clipped pixels. As in Section 4.1.2, we use the same conventional 24 bits per pixel LDR color images for our tests. Again, we generate the clipped images by clipping the *R*, *G*, *B* values that are greater than a threshold (e.g., 255×0.8 for 8 bits per color channel images). Then, we enhance the clipped images using the XDN II algorithm as well as the ZB and XDN I algorithms. To assess the quality of the corrected images, we compute the peak signal-to-noise ratio (PSNR) values (averaged over *R*, *G*, and *B* channels) of each test image for 1) the clipped image with no correction, the enhanced images generated by 2) the ZB algorithm, 3) the XDN I algorithm, and 4) the XDN II algorithm. We also use two commonly used quality metrics, the CIELAB ΔE [101] and S-CIELAB [102], to evaluate the above mentioned

three situations. For each image, the ΔE or S-CIELAB metric is the averaged value over the saturated pixels in that image.

The image-quality comparison is listed in Table 4.2. From the table, we can see that while all of the ZB, XDN I, and XDN II algorithms improve quality, the XDN II algorithm outperforms the ZB algorithm by an average of 3.86 dB in PSNR, 3.57 in CIELAB ΔE, and 0.72 in S-CIELAB, and the XDN II algorithm outperforms the XDN I algorithm by an average of 1.25 dB in PSNR, 1.09 in CIELAB ΔE, and 0.26 in S-CIELAB over all test images. The XDN II algorithm performs well especially for images with large portion of clipped areas, such as sunset, and parrots images.

Table 4.2: Objective quality comparison among the ZB, XDN I, and XDN II algorithms

Image	PSNR (in dB)				CIELAB ΔE				S-CIELAB			
	Clipped	ZB	XDN I	XDN II	Clipped	ZB	XDN I	XDN II	Clipped	ZB	XDN I	XDN II
girl	42.23	39.70	48.54	50.68	6.68	9.39	2.82	2.51	0.98	1.51	0.40	0.33
landscape	25.66	28.69	31.73	29.53	11.79	9.27	6.19	8.39	1.87	1.41	0.87	1.29
baby_girl	32.15	29.06	38.87	36.40	8.40	11.39	3.27	4.92	1.24	1.86	0.43	0.68
mountain	29.98	34.27	40.81	37.02	11.52	5.87	2.81	4.18	1.65	0.91	0.30	0.53
shoes	25.34	32.93	32.00	32.44	13.03	6.32	5.36	5.66	2.00	1.02	0.82	0.77
sunset	21.73	20.76	25.70	27.18	18.62	16.96	11.72	10.06	3.04	3.47	2.02	1.72
kodim03 (caps)	34.34	35.24	36.62	39.71	12.37	12.25	9.83	6.47	2.30	2.25	1.82	1.07
kodim05 (motorcycles)	33.62	35.74	36.50	37.07	13.46	11.00	10.79	8.20	2.25	1.83	1.77	1.17
kodim06 (boat)	25.22	28.22	26.12	32.51	17.45	10.26	14.46	7.69	3.44	1.97	2.89	1.56
kodim12 (beach)	28.41	33.65	31.76	33.16	11.04	4.55	3.99	5.98	2.08	0.81	0.74	1.08
kodim16 (lake)	35.07	35.85	36.51	41.74	11.36	10.42	8.36	5.02	2.16	1.98	1.54	0.89
kodim21 (lighthouse)	32.40	33.56	34.24	36.76	16.43	13.35	11.72	8.46	3.03	2.51	2.22	1.49
kodim23 (parrots)	29.63	31.16	33.42	34.85	10.38	8.54	6.15	5.72	1.70	1.40	1.01	0.91
Average	**30.44**	**32.22**	**34.83**	**36.08**	**12.50**	**9.97**	**7.50**	**6.40**	**2.14**	**1.76**	**1.30**	**1.04**

Figure 4.12 shows the resulting images and is used for evaluating the subjective quality of the enhanced clipped pixels and in turn the overall image. For each image, we show (in reading order) the original image, clipped image, clipped areas superimposed on the image luma, enhanced image using the ZB algorithm, and enhanced image using the XDN II algorithm. Pixel values of images in each group are linearly scaled using the same scaling factor to realize the maximum display contrast. From Figure 4.12 we can see that all clipped images have color distortions due to over exposure. The ZB algorithm corrects color for most clipped regions. However, it over-corrects the color in some clipped regions and results in further color distortion. An over-correction example can be seen in the background area of the "baby_girl" image. These artifacts happen when the color properties of the clipped region are different from the statistical properties of the unclipped regions in the image. Distortion usually occurs when the images do not possess much color variety or a large portion of clipped pixels exist. Compared to the ZB algorithm, our XDN II algorithm gives comparable or better subjective quality, without notable artifacts.

Our XDN II method works well when a saturated region is associated with an unsaturated surrounding region with similar chroma. In some cases, no such surrounding region can be found, and our method cannot be used to estimate the chroma in the clipped region. These cases are extremely difficult to handle due to lack of useful information. A possible solution is to use a classifier, as developed in [82], to classify these clipped regions as lights, reflections, or diffuse surfaces. Then, the brightness of each class of objects is enhanced by a multiplicative factor. The classifier, however, usually requires

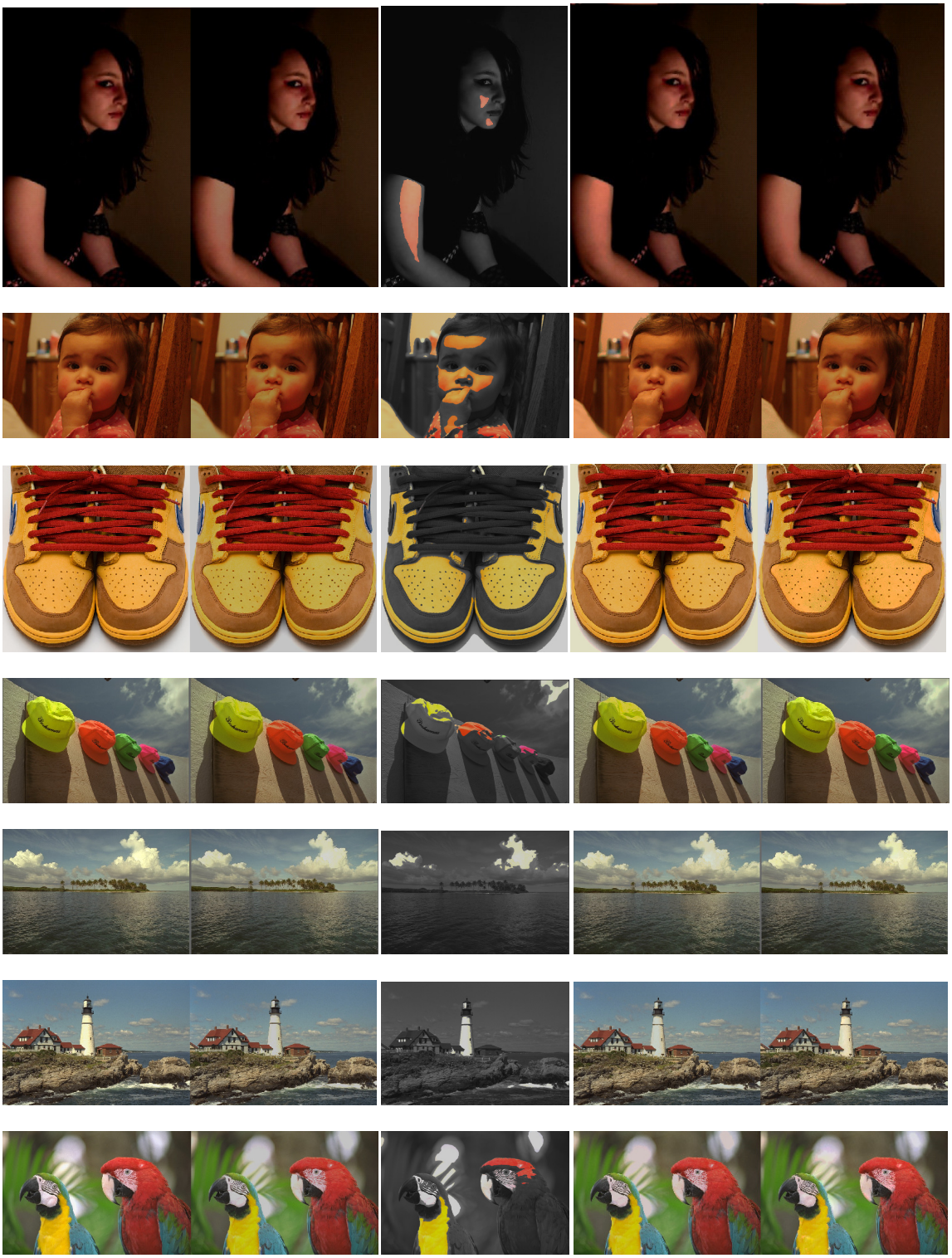
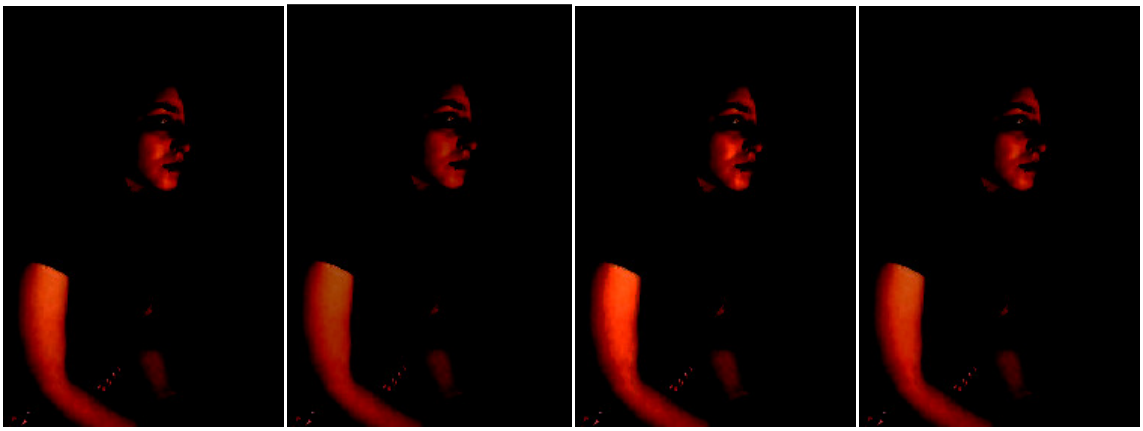


Figure 4.12: Results of clipped pixel enhancement using the XDN II algorithm. For each row, we show (from left to right) the original image, clipped image, clipped areas superimposed on the image luma, enhanced image using the ZB algorithm, and enhanced image using the XDN II algorithm.

human interaction. Furthermore, the multiplicative factors are set to rather arbitrary values (1.5 for lights, 1.25 for reflections).

One important application of the clipped-pixel color enhancement is to use it as a pre-processing step of an inverse tone mapping for producing high quality HDR images/video from existing LDR images/video. Since color clipping is often more perceptible in high-contrast inverse-tone-mapped HDR images, correcting the clipping appears more important for generating and displaying HDR images. In order to verify the importance of color correction for the clipped pixels in the contrast enhancement process, we apply an inverse tone mapping to convert LDR images into HDR images. Since a logarithmic function is the empirical model of the tone-mapping operators, we use the inverse of the logarithmic function as the inverse tone mapping operator to expand the contrast of the LDR images. The subjective quality of the inverse-tone-mapped HDR images cannot be directly shown on a conventional computer screen or print, due to the limited dynamic range of such media. For this reason, we present two "virtual" exposures of each HDR image, as done in [81], [86], to display HDR images in print. Each exposure reveals different brightness ranges of the entire dynamic range of the HDR image. Figure 4.13 shows two virtual exposures of the HDR "girl" and "baby_girl" image sets. Each column of images corresponds to (from left to right) the original unclipped image, the clipped image, the enhanced image produced by the ZB algorithm, and the enhanced image obtained by the XDN II algorithm, respectively. From Figure 4.13, especially the low exposure images, we observe that the image corrected with the XDN II approach is more similar to the original than either the clipped image or the result of the ZB algorithm.

This verifies that our color enhancement algorithm produces high quality HDR images from clipped LDR images.



Low exposure images of "girl"

High exposure images of "girl"

Low exposure images of "baby_girl"

High exposure images of "baby_girl"

Figure 4.13: Two virtual exposures of HDR images. The four columns of images correspond to (from left to right) the original unclipped LDR image, the clipped LDR image, the enhanced image obtained by the ZB algorithm, and the enhanced image produced by the XDN II algorithm, respectively.

We have also tested our algorithm on several 2D and 3D video sequences. Results show that the XDN II algorithm produces pleasant and consistent enhanced color over time for video sequences. 3D effects also look more realistic with the desaturated color. A representative frame of a 3D video sequence before and after enhancement is shown in Figure 4.14. The original left and right views of a stereoscopic video frame are shown in

Figure 4.14: An example of clipped pixel enhancement for stereoscopic 3D videos. (a) The original left and right views of a representative 3D video frame; (b) the saturated views; (c) the individually corrected left and right views; (d) the result after color-consistency adjustment between the left and right views; (e) a blow up region in (c), where color inconsistency between left and right views occurs; (f) the corresponding blow up region in (d), where color appears to be consistent between the two views.

Figure 4.14 part (a). The saturated views are given in part (b). Part (c) depicts the color correction results when applying the XDN II algorithm to the left and right views independently. Color inconsistencies between the left and right views are observed especially in the highlighted region. Part (d) shows the final results after fixing color inconsistency. The highlighted regions in parts (c) and (d) are enlarged in (e) and (f), respectively.

Subjective tests on the quality of the enhanced 3D videos have been conducted on a 46" 3D LED display (Hyundai S465D). The viewing conditions for the subjective assessment were set up based on Section 2.1 of the ITU-R BT.500-11 [43]. Twenty non-expert viewers participated in the subjective test. They were asked to rate the quality of five over-exposed 3D video sequences and the enhanced sequences. Statistical results indicate that a 10.25% quality improvement is achieved by the color enhancement over all test sequences. This is a great performance considering that the area of the saturated regions is usually a small portion of the entire video.

4.2.5 Conclusions

In this section, we have proposed an effective method for enhancing clipped pixels in 2D/3D color images and videos. We take advantage of the strong correlation between the chroma of the clipped pixels and their surrounding unclipped pixels. Our XDN II method greatly reduces the color distortion caused by clipping. It also effectively corrects the luma of pixels in the clipped areas. The XDN II method outperforms the ZB and XDN I algorithms, respectively, by an average of 3.86 dB and 1.25 dB in PSNR, 3.57 and 1.09 in CIELAB ΔE, and 0.72 and 0.26 in S-CIELAB. Subjective results also show that the

enhanced content generated by the XDN II method is visually more plausible than the clipped images and videos for both 2D and 3D cases. We show that by applying inverse tone mapping to LDR content that have been enhanced by the XDN II, we obtain more plausible and realistic HDR images than applying inverse tone mapping directly to the clipped content.

5 Conclusions and Future Work

5.1 Significance and Potential Applications of the Research

3D video can give the viewers real-life experience by providing the impression of depth. 3D technology has not yet been widely adopted due to many challenging issues that are not present in 2D imaging systems, ranging from capturing, compression, transmission, to post-processing and display. In this thesis, we address three important issues on capturing and post-processing of 3D content that significantly increase the 3D quality of experience. In particular, we provide 3D capturing and displaying guidelines in Chapter 2. In Chapter 3, we present a content-aware automatic 3D reframing technology that customizes 3D content for displays of different aspect ratios. Chapter 4 provides two algorithms for correcting color distortion caused by saturation present in LDR content. Combining the 3D and HDR capabilities provides an immersive and true to life viewing experience.

The comprehensive benchmark 3D database we provide in Chapter 2 may help researchers conduct other subjective tests to advance the emerging field of 3D technology. The 3D capturing guidelines we propose provide professional and amateur stereographers with useful rules for setting capturing parameters in order to consistently obtain high quality 3D content. These guidelines may improve the 3D viewing experience by eliminating effects that cause headache, nausea, and visual fatigue. As such, our work has the potential to boost the wide adoption of 3D technology and devices such as stereoscopic cameras and 3D displays. These guidelines can be integrated into stereoscopic camcorders, so that a warning sign will automatically show in real time on

the camcorder screen when the setting results in bad quality. Furthermore, we can significantly reduce the post-processing procedures and improve the quality of the resulting 3D images and videos if the pre-processed content is properly captured. The horizontal disparity adjustment is justified to be simple and effective. It avoids the window violation issue and brings important objects towards the comfort zone. Since the disparity adjustment is performed through digital processing, it is inexpensive to implement on the chips used in 3D camcorders. This provides high quality 3D viewing experience in real time for preview or play back on 3D capturing devices.

The 3D visual attention model is more challenging than that of 2D due to the presence of depth. Our 3D visual attention model proposed in Chapter 3 effectively identifies the prominent regions in a stereoscopic 3D video frame. It can be used in creating a 3D quality metric, where we assign more weight to distortions in the visually important regions. A good 3D quality metric will eliminate the need of extensive subjective tests, allow better real-time estimation of quality on capturing devices, more effective compression, and real-time adjustment of 3D content at the receiver end.

Our proposed automatic 3D reframing approach increases the 3D viewing experience on devices of various aspect ratios. It reduces the chance of having important objects being cropped or located at the boundary of a frame as well as experiencing window violation. The automatic reframing eliminates the need of high cost and labour intensive pan and scan. The 3D content can be pre-processed for common aspect ratios or post-processed on the fly by implementing the algorithm on chips of the 3D displays.

The color correction algorithms proposed in Chapter 4 take advantage of the HDR displays by showing higher contrast and more vivid colors, while avoiding annoying color distortions. By combining 3D with HDR technology, we produce realistic videos that resemble the real-life viewing experience. Even with conventional LDR displays, the corrected content may bring better quality of experience by removing the color distortion caused by saturation. Many other research areas can benefit from the color correction for clipped pixels. Since color has been widely used in machine-based vision systems, our algorithm may also help increase the performance of tasks such as color-based video segmentation, object recognition, tracking, panoramic video generation, and multi-view video processing.

5.2 Summary of Contributions

This thesis addresses three important issues on capturing and post-processing of stereoscopic 3D content in order to improve the 3D quality of experience. More specifically, we 1) provide capturing and disparity-adjustment guidelines for 3D images and videos, 2) design a content-aware 3D reframing algorithm to automatically adjust content for displays of different aspect ratios than the original stream, and 3) propose two methods for correcting color saturation for 2D and 3D images and videos.

- We build a 3D image and video database with the content captured using various capturing parameters, such as the lighting conditions, distances between the camera lenses to the closest object, to the furthest object, and to the object of interest. Instead of using artificial lab settings as in other databases, our database features realistic context, such as people and objects in ordinary surroundings,

which resembles content that is actually being shown on 3D broadcasting channels.

- We conduct comprehensive subjective tests to determine the influence of a few capturing parameters to the quality of 3D images and videos before and after horizontal parallax adjustment. The subjective tests are systematically conducted on 3D TVs and 3D mobile devices of different sizes. We quantitively evaluate the effect of each capturing parameter and the horizontal parallax adjustment. A set of guidelines for capturing and displaying 3D images and videos are given, based on the findings of our subjective tests, for an improved 3D quality of experience.

- We propose a novel and complete pipeline for smart content-aware 3D video reframing. This solution allows us to display high quality stereoscopic content on 3D displays with different aspect ratios than the one chosen for the original content.

- We propose a bottom-up 3D visual attention model that identifies the prominent regions in a stereoscopic 3D video frame. The model intelligently combines disparity, edges, motion, luminance, and chrominance information to generate an accurate saliency map for 3D content.

- We develop an automatic reframing approach, which creates a bounding box for each frame based on the saliency maps of the current frame and the neighboring frames. Special attention is paid to avoid the important objects from being cropped or located right at the border of the new window. In addition, the

temporal jerkiness of the cropping window is avoided by setting a location threshold and giving high priority to the bounding box location of the previous frame. A temporal low-pass filter is also employed to the bounding box locations in order to further ensure the temporal smoothness of the reframed video. Experimental results show that our scheme is very effective and robust for a great variety of stereoscopic video sequences. Our algorithm is computationally efficient and easy to implement in real time.

- We propose a fast and improved Bayesian algorithm for correcting saturation in color images. Our method utilizes images' strong spatial correlation in addition to the correlations between the R, G, and B color channels. While the state-of-the-art method have used statistics of all unsaturated pixels in an image to correct the clipped regions, we use a dilation operation to find a surrounding area for each clipped region in the image, and use statistics calculated based on this surrounding region to correct a clipped region. Experimental results show that our proposed method effectively corrects the saturated color images, and outperforms the state-of-the-art algorithm in both objective and subjective image qualities.

- We propose another effective method for enhancing clipped color pixels. The method works for images as well as videos, and is suitable for correcting both 2D and 3D content. We take advantage of the strong correlation between the chroma of the clipped pixels and their surrounding unclipped pixels. Our method greatly reduces the color distortion caused by clipping. It also effectively corrects the luma of pixels in the clipped areas. Our method significantly outperforms the

state-of-the-art algorithms based on three popular objective quality metrics. Subjective results show that the enhanced content generated by our method is visually more plausible than the clipped images and videos for both 2D and 3D cases. By applying an inverse tone mapping to the enhanced content, we obtain very plausible and realistic high-dynamic-range content that resembles the real-world scenes.

5.3 Directions for Future Work

In Chapter 2 of this thesis, we have implemented and verified a simple horizontal parallax adjustment method, which repositions the closest object on the plane of the display. This simple approach avoids window violation and provides decent quality of experience. However, while all negative disparities are eliminated, the background objects may result in too large positive disparity, which causes eye divergence. Future work could develop a smart parallax adjustment method to achieve high quality of viewing experience by avoiding window violation and eye divergence at the same time. The method could incorporate the 3D visual attention model developed in Chapter 3 in order to position the prominent part of a scene in the comfort zone [30] as well as make necessary trade-offs based on the saliency map. The sizes and aspect ratios of 3D displays could also be taken into consideration when developing the parallax adjustment algorithm.

Although it is now common to use short video sequences for subjective quality tests, using longer video sequences may offer more insightful conclusions. By doing this, the

viewers will be able to provide more useful information about the quality of their viewing experience, especially regarding visual fatigue.

In order to use the full display resolution and avoid viewing distortions such as video stretching or squeezing when 3D content is viewed on displays of different aspect ratios, we have developed an effective content-aware 3D reframing algorithm that smartly crop the video frame, leaving the important regions in the scene. The cropping is always performed on either the horizontal or vertical sides of a frame, depending on the original and targeting aspect ratios. The other two sides are unchanged to retain the maximum original content. When images and videos originally prepared for large screens are showed on small devices, sometimes it is useful to scale the prominent regions for the content to be well legible. This requires extending the reframing algorithm to perform retargeting task by cropping all four sides of a frame and then scaling the remaining area. Choosing the size and location of a small important area could be done based on the energy distribution of the saliency maps of the current frame and the neighbouring frames. Temporal smoothness of the bounding-box sizes and locations should be carefully ensured as in the proposed reframing algorithm.

Another direction for future work would be developing objective quality assessment metrics for 3D and 3D HDR content. This reduces the use of tedious subjective tests, and is very helpful in the development of 3D and 3D HDR technologies, especially when quality needs to be assessed in real time. A perceptually driven quality metric for 3D and 3D HDR videos can be developed based on existing quality metrics, such as [112], [113], and HDR-VDP-2 [26].

When displaying conventional LDR videos on HDR displays, the increase of brightness and contrast often causes false-contour artifacts, which is also called banding or posterization. Visible false contour lines are orthogonal to image-gradient directions, and often appear in smooth gradient regions. These regions require more color or intensity levels to describe them. Future work could develop methods for removing the contour artifacts while preserving the video details for inverse-tone-mapped HDR content. Contours may be removed by applying spatial smoothing. To effectively smooth contour artifacts without introducing excessive blur to an image, the contour scales in different regions need to be identified. Then, different amounts of smoothing could be applied based on the estimated contour scales.

The scales of contours depend on factors such as the background luminance, local contrast, spatial frequency, LDR compression method, and LDR quantization. Inverse-tone-mapping method affects the background luminance and local contrast of HDR content. The increase of luminance and contrast greatly raises the eye sensitivity to contours in the inverse-tone-mapped HDR content. As a result, inverse tone mapping (ITM) changes the scale distributions of false contours, and should be considered in order to accurately detect and remove the contouring artifacts in the HDR content. False-contour removing methods should be adaptive to various ITM curves, which has not been considered by the existing contour-removal methods [114], [115], [116]. In addition, future work could study the quantization artifacts introduced to the color components when the captured analog signal is converted to digital and the R, G, B channels are independently represented with limited bit depth. Such a problem is overlooked in all of the existing post-processing methods. Correction of these quantization artifacts should

result in better spatial color consistency and therefore the removal of related color contours. One possible solution is to apply decontouring in a proper color space rather than in the R, G, B channels separately.

References

[1] A. Vetro, A. M. Tourapis, K. Müller, and T. Chen, "3D-TV content storage and transmission," *IEEE Trans. Broadcast.*, Jun. 2011.

[2] C. L. Zitnick, S. B. Kang, M. Uyttendaele, S. Winder, and R. Szeliski, "High quality video view interpolation using a layered representation," in *Proc. ACM SIGGRAPH* and *ACM Trans. Graphics*, Los Angeles, CA, Aug. 2004.

[3] D. Scharstein and R. Szeliski. "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *International Journal of Computer Vision*, vol. 47, pp. 7-42, 2002.

[4] S. M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski, "A comparison and evaluation of multiview stereo reconstruction algorithms," in *International Conference on Computer Vision and Pattern Recognition*, pp. 519–528, 2006.

[5] P. Kauff, N. Atzpadin, C. Fehn, M. Müller, O. Schreer, A. Smolic, and R. Tanger, "Depth map creation and image based rendering for advanced 3DTV services providing interoperability and scalability", *Signal Processing: Image Communication.* Special Issue on 3DTV, February 2007.

[6] W. J. Tam and L. Zhang, "3D-TV content generation: 2D-to-3D conversion," *IEEE International Conference on Multimedia and Expo*, pp. 1869-1872, 2006.

[7] M. T. Pourazad, P. Nasiopoulos, and R. K. Ward, "An H.264-based scheme for 2D to 3D video conversion", *IEEE Transactions on Consumer Electronics*, vo1.55, no.2, May, 2009.

[8] M. Kawakita, T. Kurita, H. Kikuchi, and S. Inoue, "HDTV AXI-vision camera," in *Proc. International Broadcasting Conference*, pp. 397–404, 2002.

[9] G. J. Iddan and G. Yahav, "3D imaging in the studio (and elsewhere …)," in *Proc. of SPUE Videometrics and Optical Methods for 3D Shape Measurements*, pp. 48–55, 2001.

[10] M. N. Do, Q. H. Nguyen, H. T. Nguyen, D. Kubacki, and S. J. Patel, "Immersive visual communication," *Signal Processing Magazine*, *IEEE*, vol. 28, no. 1, pp. 58-66, Jan. 2011.

[11] H. Urey, K. V. Chellappan, E. Erden, and P. Surman, "State of the art in stereoscopic and autostereoscopic displays," *Proceedings of the IEEE*, vol. 99, no. 4, pp. 540-555, Apr. 2011.

[12] A. J. Woods and C. R. Harris, “Comparing levels of crosstalk with red/cyan, blue/yellow, and green/magenta anaglyph 3D glasses,” In *Proceedings of SPIE Stereoscopic Displays and Applications 2010*, vol. 7524, 2010.

[13] A. J. Woods and T. Rourke, “Ghosting in anaglyphic stereoscopic images,” in *Proc. SPIE--Stereoscopic Displays and Virtual Reality Systems XI*, Edited by A. J. Woods, J. O. Merritt, S. A. Benton, and M. T. Bolas, vol. 5291, pp. 354–365, 2004.

[14] A. J. Woods and T. Rourke, “Ghosting in anaglyphic stereoscopic images,” in *Proc. Conf. Stereoscopic Displays Virtual Reality Syst.,* San Jose, CA, Jan. 19–22, 2004.

[15] H. Jorke, “Device for projecting a stereo color image,” US Patent No. 7,001,021, Feb. 21, 2006.

[16] M. Richards and G. D. Gomes, "Spectral separation filters for 3D stereoscopic D-Cinema presentation,” US Patent Application No. 2011/0205494, Aug. 2011.

[17] L. Lipton, "Stereoscopic motion picture projection system," US Patent No. 5,481,321, Jan. 2, 1996.

[18] S. M. Faris, “Novel 3D stereoscopic imaging technology,’’ *Proc. SPIE--Int. Soc. Opt. Eng.*, vol. 2177, no. 1, pp. 180–195, 1994.

[19] C. Van Berkel, D. W. Parker, and A. R. Franklin, “Multiview 3D-LCD,” in *Proc. SPIE Stereoscopic Displays and Virtual Reality Systems*, vol. 2653, 1996, pp. 32–39.

[20] R. B. Johnson and G. A. Jacobsen, “Advances in lenticular lens arrays for visual display,’’ *Proc. SPIE--Int. Soc. Opt. Eng.,* vol. 5874, no. 1, 2005.

[21] C.-H. Tsai, P. Lai, K. Lee, and C.-K. Lee, “Fabrication of a large F-number lenticular plate and its use as a small-angle flat-top diffuser in autostereoscopic display screens,’’ *Proc. SPIE--Int. Soc. Opt. Eng.,* vol. 3957, no. 1, pp. 322–329, 2000.

[22] Scott Daly, “The visible differences predictor: An algorithm for the assessment of image fidelity,” in *Digital Images and Human Vision*, Andrew B. Watson (Ed.), MIT Press, Cambridge, MA, USA, pp. 179-206, 1993.

[23] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, “Image quality assessment: from error visibility to structural similarity,” *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600-612, Apr. 2004.

[24] L. Zhang, L. Zhang, X. Mou, and D. Zhang, “FSIM: A feature similarity index for image quality assessment,” *IEEE Transactions on Image Processing*, vol. 20, no. 8, pp. 2378-2386, Aug. 2011.

[25] R. Mantiuk, S. Daly, K. Myszkowski, and H. Seidel, "Predicting visible differences in high dynamic range images: model and its calibration," In *Proc. SPIE*, vol. 5666, 204–214, 2005.

[26] Rafał Mantiuk, Kil Joong Kim, Allan G. Rempel, and Wolfgang Heidrich, "HDR-VDP-2: A calibrated visual metric for visibility and quality predictions in all luminance conditions," *ACM Transactions on Graphics (Proc. of SIGGRAPH'11),* 30(4), article no. 40, 2011.

[27] Philipp Merkle., Karsten Müller., and Thomas Wiegand, "3D video: acquisition, coding, and display," *IEEE Transactions on Consumer Electronics*, vol. 56, no. 2, pp. 946-950, May 2010.

[28] Gerhard P. Herbig, "The three golden rules of stereography (in German)," *Stereo journal*, vol. 65, March 2002.

[29] F. Zilly, M. Muller, P. Eisert, and P. Kauff, "The stereoscopic analyzer — an image-based assistance tool for stereo shooting and 3D production," *IEEE International Conference on Image Processing (ICIP),* pp. 4029-4032, 26-29 Sept. 2010.

[30] Bernard Mendiburu, "3D movie making: stereoscopic digital cinema from script to screen." Elsevier - Focal Press, 2009.

[31] Robert S. Allison, "The camera convergence problem revisited," in *Stereoscopic Displays and Virtual Reality Systems XI, Proceedings of SPIE,* vol. 5291, San Jose, CA, USA, 2004, pp. 167-178.

[32] Andrew J. Woods, Tom Docherty, and Rolf Koch, "Image distortions in stereoscopic video systems," in *Stereoscopic Displays and Applications IV, Proc. SPIE*, vol. 1915, 1993, pp. 36-48.

[33] Manuel Lang, Alexander Hornung, Oliver Wang, Steven Poulakos, Aljoscha Smolic, and Markus Gross, "Nonlinear disparity mapping for stereoscopic 3D," in *ACM SIGGRAPH 2010*, Hugues Hoppe (Ed.). ACM, New York, NY, USA, Article 75, 10 pages.

[34] S. L. P. Yasakethu, W. A. C. Fernando, B. Kamolrat, and A. Kondoz, "Analyzing perceptual attributes of 3d video," *IEEE Transactions on Consumer Electronics*, vol. 55, no. 2, pp. 864-872, May 2009.

[35] S. L. P. Yasakethu, C. Hewage, W. Fernando, and A. Kondoz, "Quality analysis for 3D video using 2D video quality models," *IEEE Transactions on Consumer Electronics*, vol. 54, no. 4, pp. 1969-1976, November 2008.

[36] Lachlan D. Pockett and Marja P. Salmimaa, "Methods for improving the quality of user created stereoscopic content," *Stereoscopic Displays and Applications XIX, Proc. of SPIE-IS&T Electronic Imaging*, SPIE vol. 6803, 680306, 2008.

[37] Lutz Goldmann, Francesca De Simone, and Touradj Ebrahimi, "Impact of acquisition distortion on the quality of stereoscopic images," *International Workshop on Video Processing and Quality Metrics for Consumer Electronics*, Scottsdale, Arizona, USA, January 13-15, 2010.

[38] Jukka Hakkinen, Jari Takatalo, Markku Kilpelainen, Marja Salmimaa, and Gote Nyman, "Determining limits to avoid double vision in an autostereoscopic display: Disparity and image element width," *J. Soc. Inf. Display,* 17, 433 (2009).

[39] Monika Polonen, Marja Salmimaa, Viljakaisa Aaltonen, Jukka Hakkinen, and Jari Takatalo, "Subjective measures of presence and discomfort in viewers of color-separation-based stereoscopic cinema," *J. Soc. Inf. Display,* 17, 459 (2009).

[40] Salmimaa, Marja and Hakkinen, Jukka and Liinasuo, Marja and Jarvenpaa, Toni, "Effect of number of views to the viewing experience with autostereoscopic 3-D displays," *Journal of the Society for Information Display,* vol.17 Nr.5, 449-458 (2009).

[41] Bevan, Nigel, "Classifying and selecting UX and usability measures," *Proceedings of Meaningful Measures: Valid Useful User Experience Measurement (VUUM)*, 5th COST294-MAUSE Open Workshop, Reykjavik, Iceland, June 2008, pp. 13-18.

[42] David G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91-110, November 2004.

[43] ITU-R, "Methodology for the subjective assessment of the quality of television pictures," ITU-R, Tech. Rep. BT.500-11, 2002.

[44] S. Boslaugh and P. A. Watters, "*Statistics in a nutshell*," O'Reilly Media, Inc., 2008, pp. 151-168.

[45] R. Conrod, "Demystifying Active Format Description," Mason, OH, USA, 2008.

[46] L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 20, no. 11, pp. 1254-1259, November 1998.

[47] C. Chamaret, J. Chevet, and O. Le Meur, "Spatio-temporal combination of saliency maps and eye-tracking assessment of different strategies," in *IEEE International Conference on Image Processing (ICIP)*, Hong Kong, 2010.

[48] D. Walther and C. Koch, "Modeling attention to salient proto-objects," *Neural Networks,* vol. 19, no. 9, pp. 1395-1407, November 2006.

[49] F. Liu and M. Gleicher, "Video retargeting: automating pan and scan," in *the 14th annual ACM International Conference on Multimedia*, New York, NY, USA, 2006.

[50] C. Chamaret and O. Le Meur, "Attention-based video reframing: validation using-eye-tracking," in *19th International Conference on Pattern Recognition, ICPR*, Tampa, FL, USA, 2008.

[51] O. Le Meur, D. Le Callet, and D. Barba, "Predicting visual fixations on video based on low-level visual features," *Vision Research,* vol. 47, no. 19, pp. 2483-2498, September 2007.

[52] O. Le Meur, D. Le Callet, D. Barba, and D. Thoreau, "A coherent computational approach to model the bottom-up visual attention," *IEEE Trans. on Pattern Analysis and Machine Intelligence,* vol. 28, no. 5, pp. 802-817, May 2006.

[53] N. Ouerhani and H. Hugli, "Computing visual attention from scene depth," in *Int. Conf. Pattern Recognition*, Barcelona, 2000.

[54] A. Maki, J. O. Eklundh, and P. Norlund, "A computational model of depth-based attention," in *International Conference on Pattern Recognition*, Vienna, 1996.

[55] Y. Zhang, G. Jiang, M. Yu, and K. Chen, "Stereoscopic visual attention model for 3D video," in *Advances in Multimedia Modeling*, S. Boll, Q. Tian, L. Zhang, Z. Zhang and Y. Chen, Eds., Springer Berlin / Heidelberg, 2010, pp. 314 - 324.

[56] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision,* vol. 60, no. 2, pp. 91-110, November 2004.

[57] F. Alhwarin, D. Ristic-Durrant, and A. Gräser, "VF-SIFT: Very fast SIFT feature matching,," in *Proceedings of the 32nd DAGM Symposium on Pattern Recognition, 2010*, pp. 222-231, Springer, 2010.

[58] M. de Berg, O. Cheong, M. van Kreveld, and M. Overmars, *Computational Geometry: Algorithms and Applications,* Springer-Verlag, 2008.

[59] A. Treisman and G. Gelade, "A feature-integration theory of attention," *Cognitive Psychology,* vol. 12, no. 1, pp. 97-136, 1980.

[60] Q. Huynh-Thu, M. Barkowsky, and P. Le Callet, "The importance of visual attention in improving the 3D-TV viewing experience: overview and new perspectives," *IEEE Transactions on Broadcasting,* vol. 57, no. 2, pp. 421-431, June 2011.

[61] C. Guo, Q. Ma, and L. Zhang, "Spatio-temporal saliency detection using phase spectrum of quaternion Fourier transform," in *IEEE Conference on Computer Vision and Pattern Recognition*, Anchorage, Alaska, USA, 2008.

[62] T. A. Ell and S. J. Sangwine, "Hypercomplex Fourier transforms of color images," *IEEE Transactions in Image Processing,* vol. 16, no. 1, pp. 22-35, January 2007.

[63] S. Engel, X. Zhang, and B. Wandell, "Colour tuning in human visual cortex measured with functional magnetic resonance imaging," *Nature (London),* vol. 388, no. 6637, pp. 68-71, July 1997.

[64] S. Sangwine and N. Le Bihan, "Quaternion Toolbox for Matlab," 2005. [Online]. Available: http://qtfm.sourceforge.net.

[65] R. Mantiuk, S. Daly, and L. Kerofsky, "Display adaptive tone mapping," *ACM trans. on Graphics,* vol. 27, no. 3, p. 68, August 2008.

[66] J.A. Ferwerda, "Elements of early vision for computer graphics," *IEEE Computer Graphics and Applications*, vol. 21, no. 5, pp. 22-33, 2001.

[67] E. Reinhard, G. Ward, S. Pattanaik, and P. Debevec, *High Dynamic Range Imaging: Acquisition, Display, and Image-Based Lighting.* Published by Morgan Kaufmann, 2006.

[68] K. Myszkowski , R. Mantiuk, and G. Krawczyk, *High Dynamic Range Video.* Published by Morgan & Claypool Publishers, San Rafael, USA, 2008.

[69] F. Drago, K.Myszkowski, T. Annen, and N. Chiba, "Adaptive logarithmic mapping for displaying high contrast scenes," *Comput. Graph. Forum (Proc. of EUROGRAPHICS)*, vol. 24, no. 3, pp. 419-426, 2003.

[70] E. Reinhard, M. Stark, P. Shirley, and J. Ferwerda, "Photographic tone reproduction for digital images," *ACM Trans. Graph. (Proc. SIGGRAPH)*, vol. 21, no. 3, pp. 267-276, 2002.

[71] E. Reinhard and K. Devlin, "Dynamic range reduction inspired by photoreceptor physiology," *IEEE Trans. on Visual. Comput. Graph.*, vol. 11, no. 1, pp. 13-24, 2005.

[72] G. Ward, H. Rushmeier, and C. Piatko, "A visibility matching tone reproduction operator for high dynamic range scenes," *IEEE Trans. Visual. Comput. Graph.*, vol. 3, no. 4, pp. 291-306, 1997.

[73] G. Krawczyk, K. Myszkowski, and H.-P. Seidel, "Lightness perception in tone reproduction for high dynamic range images," *Comput. Graph. Forum (Proc.EUROGRAPHICS)*, vol. 24, no. 3, pp. 635-645, 2005.

[74] R. Fattal, D. Lischinski, and M. Werman, "Gradient domain high dynamic range compression," *ACM Trans. Graph. (Proc. SIGGRAPH)*, vol. 21, no. 3, pp. 249-256, 2002.

[75] R. Mantiuk, K. Myszkowski, and H.-P. Seidel, "A perceptual framework for contrast processing of high dynamic range images," *ACM Trans. Appli. Percept.*, vol. 3, no. 3, pp. 286–308, 2006.

[76] H. Seetzen, W. Heidrich, W. Stuerzlinger, G. Ward, L. Whitehead, M. Trentacoste, A. Ghosh, and A. Vorozcovs, "High dynamic range display systems," *ACM Transactions on Graphics*, vol. 23, no. 3, pp. 760-768, August 2004.

[77] O.A. Akyüz, R. Fleming, B.E. Riecke, E. Reinhard, and H.H. Bülthoff, "Do HDR displays support LDR content? A psychophysical evaluation," *ACM Transactions on Graphics*, vol. 26, no. 3, Jul. 2007.

[78] L. Meylan, S. Daly, and S. Süsstrunk, "The reproduction of specular highlights on high dynamic range displays," *In Proc. of the 14th Color Imaging Conference*, 2006.

[79] L. Meylan, S. Daly, and S. Süsstrunk, "Tone mapping for high dynamic range displays," *In Proc. IS&T/SPIE Electronic Imaging: Human Vision and Electronic Imaging XII,* vol. 6492, 2007.

[80] F. Banterle, P. Ledda, K. Debattista, and A. Chalmers, "Inverse tone mapping," In *Proceedings of the 4th International Conference on Computer Graphics and Interactive Techniques in Australasia and Southeast Asia* (Kuala Lumpur, Malaysia, November 29 - December 02, 2006). GRAPHITE '06. ACM, New York, NY, pp. 349-356.

[81] A.G. Rempel, M. Trentacoste, H. Seetzen, H.D. Young, W. Heidrich, L. Whitehead, and G. Ward, "LDR2HDR: On-the-fly reverse tone mapping of legacy video and photographs," *ACM Transactions on Graphics (Proc. SIGGRAPH)*, vol. 26, no. 3, 2007.

[82] P. Didyk, R. Mantiuk, M. Hein, and H.-P. Seidel, "Enhancement of bright video features for HDR displays," *Comput. Graph. Forum*, vol.27, no.4, pp. 1265-1274, 2008.

[83] G. J. Klinker, S. A. Shafer, and T. Kanade, "The measurement of highlights in color images," *International Journal of Computer Vision*, vol. 2, pp. 7-32, 1988.

[84] H.-L. Shen and Q.-Y. Cai, "Simple and efficient method for specularity removal in an image," *Applied Optics*, vol. 48, no. 14, pp. 2711-2719, 2009.

[85] P. Tan, S. Lin, L. Quan, and H.-Y. Shum, "Highlight removal by illumination-constrained inpainting," *In Ninth IEEE International Conference on Computer Vision*, pp. 164-169, 2003.

[86] L. Wang, L.-Y. Wei, K. Zhou, B. Guo, and H.-Y. Shum, "High dynamic range image hallucination," *International Conference on Computer Graphics and Interactive Techniques*, ACM SIGGRAPH, no.72, San Diego, California, 2007.

[87] X. Zhang and D. H. Brainard, "Estimation of saturated pixel values in digital color imaging," *Journal of the Optical Society of America A*, Optical Society of America, 2004, vol. 21, no. 12, pp. 2301-2310.

[88] B. K. Gunturk, J. Glotzbach, Y. Altunbasak, R. M. Mersereau, and R. W. Schafer, "Demosaicking: color filter array interpolation," *IEEE Signal Processing Mag.*, vol. 22, no. 1, pp. 44-54, 2005.

[89] S. Farsiu, M. Elad, and P. Milanfar, "Multiframe demosaicing and super-resolution of color images," *IEEE Transactions on Image Processing*, vol. 15, no. 1, pp. 141-159, 2006.

[90] J. E. Adams and J. F. Hamilton Jr., "Adaptive color plane interpolation in single color electronic camera," U.S. Patent 5 506 619, Apr. 1996.

[91] Kodak Lossless True Color Image Suite, available at http://r0k.us/graphics/kodak/.

[92] H. Landis, "Production-ready global illumination," *In Siggraph Course Notes 16*, 2002.

[93] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," *Proceedings of the 1998 IEEE International Conference on Computer Vision*, Bombay, India.

[94] F. Durand and J. Dorsey, "Fast bilateral filtering for the display of high-dynamic-range images," *ACM Transactions on Graphics (TOG)*, vol. 21, no. 3, July 2002.

[95] S. Paris and F. Durand, "A fast approximation of the bilateral filter using a signal processing approach," *In Proc. of Eur. Conf. on Comp. Vision.*, 2006.

[96] W. R. Dillon and M. Goldstein, *Multivariate Analysis.* Wiley, New York, 1984.

[97] R. C. Gonzalez and R. E. Woods, *Digital Image Processing.* Addison-Wesley Longman Publishing Co., Inc., Boston, MA, 1992.

[98] H. Knutsson and C.-F. Westin, "Normalized and differential convolution: methods for interpolation and filtering of incomplete and uncertain data," *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 515-523, New York City, USA, Jun. 1993.

[99] ITU-R, Rec. BT. 601-4, Encoding Parameters of Digital Television for Studios.

[100] J.E. Gentle, *Cholesky Factorization in Numerical Linear Algebra for Applications in Statistics.* Berlin: Springer-Verlag, pp. 93-95, 1998.

[101] CIE (Commission Internationale de l'Eclairage). Colorimetry technical report. CIE Pub. No.15, 2nd ed. Vienna, Austria: Bureau Central de la CIE, 1986, [corrected reprint 1996].

[102] X. Zhang and B.A. Wandell, "A spatial extension of CIELAB for digital color-image reproduction," *Journal of the Society for Information Display*, vol 5, no. 1, pp. 61-63, Mar. 1997.

[103] R. Mantiuk, S. Daly, and L. Kerofsky, "Display adaptive tone mapping," *ACM trans. on Graphics*, vol. 27, no. 3, pp. 68, July 2008.

[104] B. Julesz, *Foundations of Cyclopean Perception.* University of Chicago Press, Chicago, IL, USA, 1971.

[105] U. Fecker, M. Barkowsky, and A. Kaup, "Improving the prediction efficiency for multi-view video coding using histogram matching," in *Proc. Picture Coding Symposium 2006*, pp. 2–16, Apr. 2006.

[106] M. Flierl, A. Mavlankar, and B. Girod, "Motion and disparity compensated coding for multiview video," *IEEE Trans. Circuits Syst. Video Technol.,* vol. 17, no. 11, pp. 1474-1484, Nov. 2007.

[107] U. Fecker, M. Barkowsky, and A. Kaup, "Histogram-based pre-filtering for luminance and chrominance compensation of multi-view video," *IEEE Trans Circuits and Systems for Video Technology*, vol. 18, no. 9, pp. 1258-1267, Sept. 2008.

[108] Y. Chen, C. Cai, and J. Liu, "YUV correction for multi-view video compression," *Proc. Int. Conf. Pattern Recognition 2006*, pp. 734-737, Aug. 2006.

[109] F. Shao, G. Jiang, M. YuI, and K. Chen, "A content-adaptive multi-view video color correction algorithm," *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP),* pp. 969-972, Apr. 2007.

[110] C. Doutre and P. Nasiopoulos, "Color correction preprocessing for multiview video coding." *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 19, no. 9, pp. 1400-1406, Sept. 2009.

[111] F. Shao, G. Jiang, and M. Yu, "A robust color correction method for stereoscopic video coding," *3rd International Congress on Image and Signal Processing (CISP),* vol. 3, pp. 1106-1109, Oct. 2010.

[112] P. Joveluro, H. Malekmohamadi, W.A.C. Fernando, and A.M. Kondoz, "Perceptual video quality metric for 3D video quality assessment," *3DTV-Conference: The True Vision - Capture, Transmission and Display of 3D Video (3DTV-CON)*, 2010, pp. 1-4, June 2010.

[113] L. Jin, A. Boev, A. Gotchev, and K. Egiazarian, "Validation of a new full reference metric for quality assessment of mobile 3DTV content", the *19th European Signal Processing Conference (EUSIPCO-2011)*, Barcelona, Spain, August 29-September 2, 2011.

[114] S. Daly and X. Feng, "Decontouring: Prevention and removal of false contour artifacts," in *Proc. Human Vision and Electronic Imaging IX, SPIE*, vol. 5292, 2004, pp. 130–149.

[115] S. Bhagavathy, J. Llach, and J. Zhai, "Mulit-scale probabilisitic dithering for suppressing banding artifacts in digital images," in *IEEE International Conf. on Image Processing (ICIP)*, vol. 4, 2007, pp. IV–397–400.

[116] C. R. Carlson, E. H. Adelson, and C. H. Anderson, "System for coring an image representing signal," in US Patent 4,523,230. United States Patent and Trademark Office, 1985.