UNIVERSITY OF CALIFORNIA, SAN DIEGO

Perceptual Video Quality Preservation and Enhancement

A dissertation submitted in partial satisfaction of the
requirements for the degree
Doctor of Philosophy

in

Electrical Engineering
(Signal and Image Processing)

by

Qing Song

Committee in charge:

Professor Pamela C. Cosman, Chair
Professor Laurence B. Milstein, Co-Chair
Professor William S. Hodgkiss
Professor Truong Q. Nguyen
Professor Geoffrey M. Voelker

2017

The dissertation of Qing Song is approved, and it is acceptable in quality and form for publication on microfilm and electronically:

_____

_____

_____

_____
Co-Chair

_____
Chair

University of California, San Diego

2017

DEDICATION

To my family.

EPIGRAPH

Ever tried. Ever failed. No matter. Try again. Fail again. Fail better.

—Samuel Beckett

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to everyone who supported me during my Ph.D. study.

First and foremost, I would like to thank my advisor, Prof. Pamela Cosman, who accepted me as her Ph.D. student and offered me her mentorship. I was a rookie in video compression and processing when I started working with her five years ago. She guided me into this area and inspired my enthusiasm. I learned a lot in this journey under her guidance. I really appreciate all her contributions in time and ideas, and her emotional and financial support. She even met with me in her house during holidays while her son was resting at home after a heart surgery. She always revised my papers precisely and patiently, from grammar to every technical detail. She helped me practice presentations and always provided helpful suggestions. I have benefited from the writing and presentation skills learned from her, and this benefit will absolutely influence me positively in the future.

I greatly appreciate my dissertation co-chair, Prof. Laurence Milstein, and committee members, Prof. Truong Nguyen, Prof. William Hodgkiss and Prof. Geoffrey Voelker, for their precious time and feedback in my Ph.D. preliminary exam, qualifying exam and final defense. Their suggestions improved my research and this dissertation. Special acknowledgment to Prof. Milstein for his direction in the two projects about video transmission on which we collaborated, though not included in this dissertation.

I am really grateful to Dr. Guan-Ming Su, my supervisor at Dolby Labs and a co-author of the two HDR papers included in this dissertation. Had he not accepted me as an intern in the summer of 2014, I would not have had the opportunity to learn this new form of video. He taught me the basis of HDR, and proposed the debanding project. In 2016, he re-hired me and allowed me to explore the debanding filter further, even though it was not the main interest of his group. His insights and expertise in video processing

and compression greatly improved my research. I enjoyed the freedom to explore the area and the discussions about every detail with Dr. Su. The experience at Dolby Labs not only inspired me with confidence, but also inspired me with the idea of ambient light adaptation which is also included in this dissertation.

My deepest gratitude goes to my family. My beloved parents in China provide enormous support and encouragement to me. I gain courage every time we talk over the phone. I felt fully charged and energized every time I returned from China after I stayed with them. I would like to specially thank my mother who is always willing to listen and discuss about every little thing in my life, happiness or anxiety, achievement or failures. She supports every decision I made, and gives great comfort to my heart. I am grateful to her sacrifice in all these years. I owe my heartfelt thanks to Congyao, my boyfriend and my best friend, for his unfailing love and care which lightens my life. His patience and understanding ease my nerves and give me strength. I could not have reached my goals without his companionship.

I would also like to gratefully acknowledge my colleagues and friends. This dissertation includes four subjective tests, and each involved 10 - 30 people. My colleagues and friends helped me to complete the experiments without asking for a return. I thank them very much for their time and efforts. This dissertation could not be completed without their kind help.

Chapter 2 of this dissertation, in part, is a reprint of material as it appears in Q. Song and P. C. Cosman, "Luminance and detail enhancement of videos adapted to ambient illumination", submitted to *IEEE Transactions on Image Processing*. The dissertation author was the primary author of this paper and the co-author Prof. Cosman supervised the research.

Chapter 3 of this dissertation, in part, is a reprint of material as it appears in Q. Song, P. C. Cosman, M. He, R. Vanam, L. J. Kerofsky, and Y. A. Reznik, "Subjective

quality of video bit-rate reduction by distance adaptation", *International Workshop on Video Processing and Quality Metrics for Consumer Electronics*, Feb. 2015. The dissertation author was the primary author and the co-author Prof. Cosman directed and supervised the research which forms the basis for Chapter 3. The co-author M. He helped with the subjective tests. The co-authors Dr. Vanam, Dr. Kerofsky and Dr. Reznik also contributed to the ideas in this work.

Chapter 4, in part, is a reprint of material as it appears in Q. Song, G.-M. Su, and P. C. Cosman, "Efficient debanding filtering for inverse tone mapped high dynamic range videos", submitted to *IEEE Transactions on Image Processing*, and Q. Song, G.-M. Su, and P. C. Cosman, "Hardware-efficient debanding and visual enhancement filter for inverse tone mapped high dynamic range images and videos", *International Conference on Image Processing*, pp. 3299-3303, Sep. 2016. The dissertation author was the primary author and the co-author Dr. Su directed and supervised the research which forms the basis for Chapter 4. The co-author Prof. Cosman also supervised this work.

Chapter 5, in part, is a reprint of material as it appears in Q. Song and P. C. Cosman, "Packet loss visibility of view+depth compressed stereo 3D video", *International Packet Video Workshop*, pp. 1-7, Dec. 2013. The dissertation author was the primary author of this paper and the co-author Prof. Cosman directed and supervised the research which forms the basis of Chapter 5.

VITA

2011	B. Eng. in Automation, Tongji University
2013	M. S. in Electrical Engineering (Signal and Image Processing), University of California, San Diego
2017	Ph. D. in Electrical Engineering (Signal and Image Processing), University of California, San Diego

PUBLICATIONS

Qing Song and Pamela C. Cosman, "Luminance and detail enhancement of videos adapted to ambient illumination", submitted to *IEEE Transactions on Image Processing*.

Qing Song, Guan-Ming Su, and Pamela C. Cosman, "Efficient debanding filtering for inverse tone mapped high dynamic range videos", submitted to *IEEE Transactions on Image Processing*.

Young-Ho Jung, Qing Song, Kyung-Ho Kim, Pamela C. Cosman and Laurence B. Milstein, "Cross-Layer Resource Allocation Using Video Slice Header Information for Wireless Transmission over LTE", accepted for publication, *IEEE Transactions on Circuits and Systems for Video Technology*.

Qing Song, Guan-Ming Su, and Pamela C. Cosman, "Hardware-efficient debanding and visual enhancement filter for inverse tone mapped high dynamic range images and videos", *International Conference on Image Processing*, pp. 3299-3303, Sep. 2016.

Qing Song, Arash Vosoughi, Pamela C. Cosman, Laurence B. Milstein, "Joint Error-Resilient Video Source Coding and FEC Code Rate Optimization for an AWGN Channel", *International Conference on Image Processing (ICIP 2016)*, pp. 2107-2111, Sep. 2016.

Qing Song, Pamela C. Cosman, Morgan He, Rahul Vanam, Louis J. Kerofsky, and Yuriy A. Reznik, "Subjective quality of video bit-rate reduction by distance adaptation", *International Workshop on Video Processing and Quality Metrics for Consumer Electronics*, Feb. 2015.

Qing Song and Pamela C. Cosman, "Packet loss visibility of view+depth compressed stereo 3D video", *International Packet Video Workshop*, pp. 1-7, Dec. 2013.

ABSTRACT OF THE DISSERTATION

Perceptual Video Quality Preservation and Enhancement

by

Qing Song

Doctor of Philosophy in Electrical Engineering
(Signal and Image Processing)

University of California, San Diego, 2017

Professor Pamela C. Cosman, Chair
Professor Laurence B. Milstein, Co-Chair

The perceptual quality of videos has attracted much attention, because it is hard to estimate by objective metrics, and because it is affected by many conditions. Compression, transmission and viewing conditions all impact the perceptual quality. In this dissertation, we aim to preserve and enhance the perceptual quality in different cases.

The impact of ambient illumination on the perceptual quality of traditional 8-bit 2D video is first studied. Some details, especially those in dark areas of videos, are invisible in bright ambient light, because of the reflection of ambient light and the

reduction of the sensitivity of human eyes. We analyze the display characteristics and human visual sensitivity, and propose methods to enhance the contrast and details without increasing the peak brightness of the display.

Another viewing condition, viewing distance, is also investigated in this dissertation. A display device held farther away may have fewer details visible compared to a device held closer. The unnoticeable details can be filtered before compression, which can reduce the bit-rate of the video. A subjective test was conducted to demonstrate the bit-rate saving without degrading the perceptual quality.

Besides the traditional 8-bit videos, a new form of video, high dynamic range (HDR) videos, is studied in Chapter 4 of this dissertation. There can be banding artifacts in the inverse tone mapped HDR videos which degrade the perceptual quality, though the impact on the objective quality is subtle. An enhancement filter is proposed to remove the banding artifacts and reduce compression artifacts, and at the same time, preserve true edges and details. The parameters of the filter are determined by minimizing the proposed perceptual distortion metric.

The perceptual quality of 3D video is explored in Chapter 5. In particular, the stereoscopic 3D video of the 2D+depth format is studied. Transmission of such videos through networks can be affected by packet losses. The importance of packets is investigated by a human observer experiment. A prediction model of the importance is developed using features such as the video type, frame type, and spatial location of the packets.

Chapter 1

Introduction

1.1 Perceptual Quality of Video

The perceptual quality of video is affected in multiple ways as the video is delivered to a viewer. Fig. 1.1 shows the pipeline of delivery. After the video is captured and produced, it is compressed by a video encoder. The output bitstream is distributed by either hard disks or networks to the viewer. The received video is displayed on the viewer selected device under the selected viewing conditions, after the bitstream is decompressed by a decoder. The decoder can be built in the selected device. In the following, we will discuss how the perceptual quality of video is affected in each step, and the possible ways to preserve and enhance the perceptual quality.

First, videos are usually compressed before distribution because of the limitation of storage and distribution. The size of one frame of a raw 8-bit video with resolution 1920×1080 and chroma subsampling 4:2:0 (i.e., the two chroma channels are downsampled by 2 in the horizontal and vertical directions) is $1920 \times 1080 \times 1.5 = 3,110,400$ bytes. Therefore, the size of a 2-hour raw video of 24 frames per second is about 500 Gigabytes. After compression, the size of the video bitstream can be reduced to just

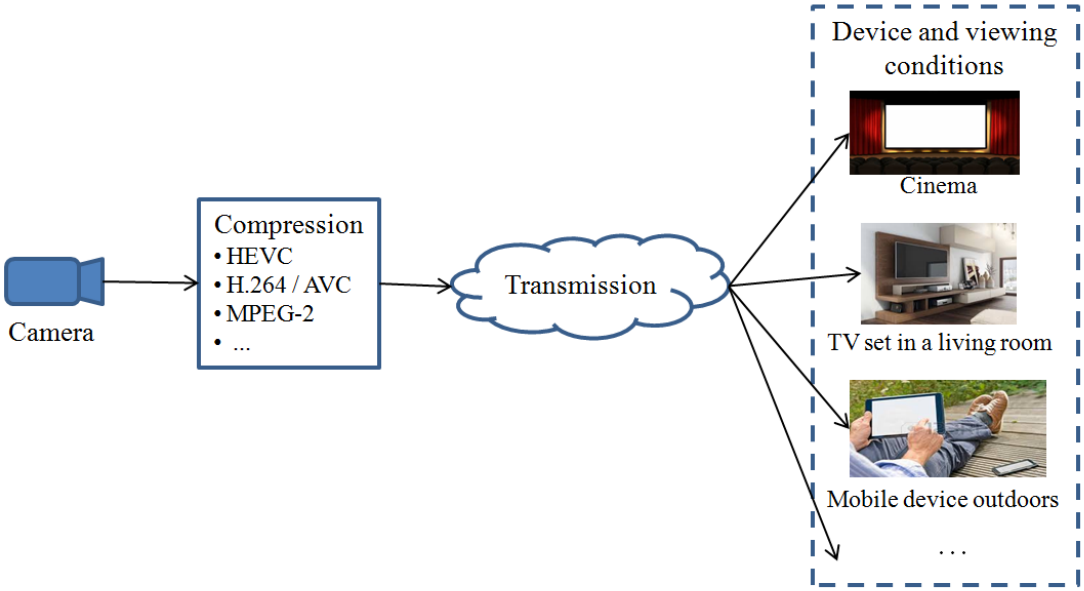


Figure 1.1: Video deliver pipeline.

several Gigabytes. Compression is usually lossy to achieve low bit-rate of the output bitstream. Lossy compression degrades the perceptual quality. A video encoder with higher compression efficiency can provide higher quality for a given bit rate. For example, the latest video compression standard HEVC [1] is more efficient than its predecessor H.264 / AVC [2]. Improving the compression efficiency has been intensively investigated for several decades.

If a video is distributed through networks, additional distortion can be caused by corruption of video packets due to bit errors, congestion, etc. To improve the quality, video packets can be protected by forward error correction. The impact of the video packets on the perceptual quality of the video can be different. For example, packets which include more motion information are usually more important, because they are more difficult to conceal when they are lost. Packets at the center of the screen can be more important, since they typically include more motion, and draw the viewer's attention. Moreover, in the recent generations of video compression standards, videos

are compressed in I, P, and B-frames. I-frames do not use any information from other frames. P-frames use previous I or P-frames as a reference. B-frames use both previous and also future I, P, or B-frames as references. Errors in different types of frames have different lengths of propagation, which affects the importance of packets. We want to protect important packets with strong protection. How to estimate the importance of video packets has been studied in many works, such as [3, 4, 5].

When the video is displayed after it is received by the viewer, the viewing conditions also play a role in the perceptual quality. Videos can be displayed on various devices under various viewing conditions: they can be watched in a dark cinema; they can be displayed on a 55" television in a living room; they can be shown on a 10" tablet outdoors. The perceptual quality of video is affected by display size and viewing distance, display brightness and ambient illumination, user movement, etc.

Among the viewing conditions, ambient illumination can greatly degrade the quality of experience. The contrast of the display is reduced by bright ambient light, and the viewer's eyes are less sensitive under bright ambient light. As a result, the video looks washed out, and many details are invisible. Often, we cannot change the ambient light. However, we can enhance the contrast and luminance of the video so that more details can be perceived. In Chapter 2, we propose two enhancement methods that improve the luminance and the visibility of details. One is content independent and thus can be applied to any video for the given device and the given ambient illumination. The other method uses simple statistics of the video content. Both methods work efficiently.

In some circumstances, some details in the videos are unnoticeable due to the behaviors of the viewer. For example, fewer details can be noticed if the viewer is in a moving car or working out on a running machine, compared to when the viewer is sitting still. The movement of the viewer greatly reduces the sensitivity of his/her eyes. For another instance, fewer details can be perceived with the increase of the viewing distance

when a video is shown on a given device, also due to the decrease of the sensitivity of eyes. In these cases, transmission of those invisible details would waste bits. In [6], a perceptual pre-filter is proposed to remove the spatial oscillations in a video that are invisible under the given viewing distance, resulting in lower complexity images which can be compressed at a lower bit-rate without loss of perceptual quality. In Chapter 3, we demonstrate the performance of the pre-filter by subjective tests. We study three viewing distances, corresponding to holding a tablet in the hand, on the lap, or on a stand. The visual quality of the compressed videos with and without the pre-filtering is compared, and we found that substantial bit-rate can be saved without degradation of perceptual quality.

1.2 New Formats of Video

The discussion above is mostly about traditional 8-bit 2D videos. There are new forms of videos attracting great interest in recent years, such as 3D and high dynamic range (HDR) videos. In addition to the aforementioned factors that affect the perceptual quality, perceptual quality of 3D and HDR videos is influenced by their own characteristics.

1.2.1 HDR Videos

HDR videos are represented in 12+ bit depth, i.e., 12 or more bits per color component [7]. They provide a wider range of brightness and a larger color gamut than the traditional 8-bit low dynamic range (LDR) videos. The pictures have higher contrast, show more details, and look truer to life than LDR videos. However, today there is limited content in HDR, because HDR displays are not widely spread yet, and the major distribution of videos is mostly at 8-bit depth. If one wants to present LDR content on a

HDR display, one has to convert the LDR content to HDR. The process is called inverse tone mapping [8].

HDR video generated by inverse tone mapping sometimes suffers from banding artifacts. The artifacts usually occur at smooth regions. They look annoying on a HDR display, and degrade the perceptual quality. In Chapter 4, we design an edge-aware selective sparse filter to generate more codewords to alleviate the banding artifacts while preserving true edges and details. Compression artifacts, such as blocky artifacts, can also be reduced by this filter. The filter works more efficiently than dense filters. Some parameters of the filter are content dependent. We propose a parameter selection mechanism which considers the smoothness and fidelity of the filtering output.

1.2.2 3D Videos

3D imagery has long been studied in the past two centuries, but 3D videos were not widely spread until the recent decade, also due to the limitation of displays and distribution.

3D videos are represented by two or more views. For the most widely used stereoscopic 3D, only two views are provided to the viewer, one for each eye. The two views are vertically aligned and horizontally slightly offset. Closer objects in the images exhibit a relatively larger offset between the two stereo images. The offsets form the disparity cue which gives the depth perception in the brain. Other 3D formats, such as free viewpoint television and multiview 3D television, provide more than two views, so that a larger scene range is presented.

From the viewpoint of compression, there is great redundancy among views. In 2009, the first stereoscopic video compression standard, multiview video coding (MVC), was released, which allows for efficient compression of 3D videos. One view is encoded as an independent 2D video which is named as *anchor*. The other view(s) is(are)

compressed using the anchor as a reference picture. Another 3D compression format is 2D+depth: the anchor is the same as that in MVC, and a depth map is compressed instead of compressing the other view(s). The depth map gives the distance between the object and the camera. The other view(s) is(are) synthesized from the anchor and the depth map at the decoder. This format saves more bit-rate than MVC.

If the 3D video is delivered by a network, video packet losses can degrade the perceptual quality. In Chapter 5, we study the importance of packets of 3D videos compressed in 2D+depth. The 2D color video is generally more important than the depth video, since the 2D video affects both views, while the depth video only affects the synthesized view. However, the importance of packets also depends on the frame type, the covered area, etc. We build a prediction model of the importance with features extracted from the video.

1.3 Subjective tests

The perceptual quality of video is difficult to estimate. Simple objective metrics, such as peak signal to noise ratio (PSNR) and the structural similarity (SSIM) index [9], do not represent the perceptual quality well. To evaluate the performance of our proposed methods, we conducted subjective tests with human observers. The human observers watched the test images / videos, and rated the quality.

There are many methods for subjective assessment, such as absolute category rating, degradation category rating, pair comparison, etc [10, 11]. Each has pros and cons. For example, pair comparison means the human observers compare images / videos in pairs, and select the preferred images / videos. It is the most direct and accurate way to compare the performances of two algorithms, but the experiment would take a long time if more than two algorithms are compared. Absolute category rating means

the test images / videos are rated independently on a category scale (e.g., 5-level scale: “excellent”, “good”, “fair”, “poor”, “bad”). It requires less time than pair comparison, but the assessment can be less accurate. In each chapter, we select the assessment method according to the purposes.

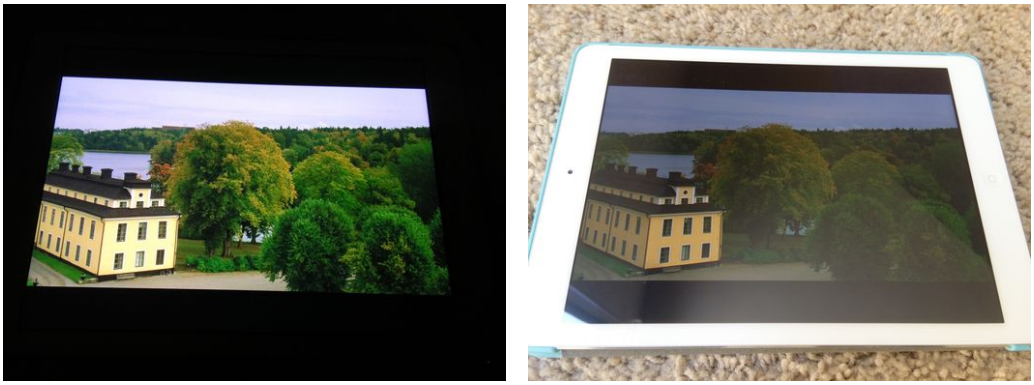
In each experiment, a description of the procedure and the opinion scale was given in written form. A training session was given to the human observers, which showed the range and type of stimuli. Each experiment took no more than 1 hour. The responses of the human observers were analyzed after the experiments.

Chapter 2

Luminance and Detail Enhancement of Videos Adapted to Ambient Illumination

In this chapter, we discuss the enhancement of video perceptual quality under bright ambient light. Among the viewing conditions discussed in Chapter 1, ambient illumination varies greatly. A typical living room is 50 lx; a bright office can be 500 lx; outdoor under shade can be 5000 lx; an overcast day can be 10,000 lx; and under direct sunlight, it can be 100,000 lx. In a room with windows, the ambient illumination can vary from 0 to 1000 lx at different times of day.

When our eyes are adapted to bright ambient light, the amount of light that enters our eyes is affected, and thus the visual sensitivity is affected [12, 13]. In addition, the reflection of ambient light reduces the contrast of a display (contrast is defined in Sec. 2.1). Therefore, fewer details can be perceived in bright surroundings than in dark. Moreover, the detail loss is more severe in dark areas in the video. Fig. 2.1 shows an image displayed in dark and in bright ambient. When displayed in the dark, the image



(a) in the dark (b) in bright ambient

Figure 2.1: An image displayed in different ambient illuminations

looks bright, and shows good details. When displayed in bright ambient light, it loses details and contrast, and appears dull and washed out. Though increasing the display brightness can compensate for some of the detail loss, it will drain the battery of the device. Note that the maximum brightness of most mobile devices today is about 400 - 600 cd/m^2 [14]. In very bright ambient light (e.g., 10,000 lx), the quality of experience of viewing the display at even the maximum brightness is still not comparable to viewing in the dark.

Some works have studied video enhancement to compensate for the effects of ambient light. Mantiuk et al. [15] constructed a tone mapping operator so that the human visual response of the enhanced image under bright ambient illumination can be as close to the maximum response as possible. The algorithm involves Laplacian decomposition and quadratic programming, and is complicated. In [16], images are enhanced by adjusting the backlight (screen brightness) to achieve the same visual response as in low ambient light. This method results in increasing the screen luminance for white. If the screen brightness for white is fixed, the method will result in clipping the bright areas of images. In [17], the tone mapping curve is constructed by establishing a linear relation between the display luminance and visual response. Kim [18] modeled an

ambient-affected contrast sensitivity function, and designed an adaptive weighting filter in the spatial frequency domain. In [19], images are enhanced by boosting the gradients and improving the brightness by linear mapping. However, contents in bright areas are clipped, and the reflection of ambient light is not considered. Su et al. [20] proposed to enhance luminance using an exponential function and to increase the gradients of the image, by taking into account both the reflection and visual sensitivity.

In this chapter, we propose two tone mapping operators to enhance the detail visibility of videos. One is content independent; the other uses some video statistics. Both need very light computation. They are built under the condition that the relationship between the amplitude level (codeword, or pixel value) and the display luminance is fixed, and the screen brightness for white is not allowed to increase. The device can detect the ambient illumination using its built-in ambient light sensor. If the computing resource of the device is very limited, the content independent tone mapping can be constructed for the given ambient illumination, and can be applied to any video. If a bit more computing resource is available, the content dependent tone mapping can be derived for each frame or group of frames, depending on how content statistics are used. No other image processing (e.g., gradient enhancement) is used.

The rest of the chapter is organized as follows: In Sec. 2.1, the display and contrast models are explained, and the proposed tone mapping operators are described in Sec. 2.2. Sec. 2.3 shows the performance of the tone mapping and the comparisons with other tone mapping methods. Sec. 2.4 summarizes the chapter.

2.1 Display and Contrast Models

2.1.1 Display Luminance

According to [21], the electro-optical transfer function (EOTF) of a given 8-bit display is:

$$\begin{aligned} L_d(Y,L_W) &= a(L_W)\big(\max[\frac{Y}{255}+b(L_W),0]\big)^{\gamma}, \\ \text{where} \quad a(L_W) &= (L_W^{\frac{1}{\gamma}}-L_B(L_W)^{\frac{1}{\gamma}})^{\gamma}, \\ b(L_W) &= \frac{L_B(L_W)^{\frac{1}{\gamma}}}{L_W^{\frac{1}{\gamma}}-L_B(L_W)^{\frac{1}{\gamma}}}, \end{aligned} \tag{2.1}$$

where L_d is the display luminance in $\mathrm{cd/m^2}$, Y is the luma value (0-255) of a pixel, γ is a display gamma, L_W is the selected screen brightness for white in $\mathrm{cd/m^2}$, and $L_B(L_W)$ is the screen brightness for black which is determined by L_W for a given device. L_W and L_B are non-negative, so (2.1) can be reduced to $L_d(Y,L_W) = a(L_W)\big(\frac{Y}{255}+b(L_W)\big)^{\gamma}$.

According to [15], the reflected light of the ambient illumination can be modeled as:

$$L_{refl}(E_{amb}) = \frac{k}{\pi}E_{amb}, \tag{2.2}$$

where k is the reflectivity of the display, and E_{amb} is the ambient illumination in lx. The total luminance from a display is:

$$\begin{aligned} L_{total}(Y,L_W,E_{amb}) &= L_d(Y,L_W)+L_{refl}(E_{amb}) \\ &= a(L_W)\big(\frac{Y}{255}+b(L_W)\big)^{\gamma}+\frac{k}{\pi}E_{amb}. \end{aligned} \tag{2.3}$$

The contrast between each two consecutive codewords (namely, codeword contrast) is

calculated as:

$$C_d(Y, L_W, E_{amb}) = 2\frac{L_{total}(Y+1, L_W, E_{amb}) - L_{total}(Y, L_W, E_{amb})}{L_{total}(Y+1, L_W, E_{amb}) + L_{total}(Y, L_W, E_{amb})}, \tag{2.4}$$

for $Y = 0, 1, \cdots, 254$. Fig. 2.3a shows the 8-bit display codeword contrast (dash red curve) when there is no reflected light from ambient illumination ($L_{refl} = 0$). The maximum screen (L_W) is set to 100 cd/m^2. The display gamma is 2.23, and the reflectivity is 6.5%, which are the values for an iPad Air from the Display Mate website [14].

2.1.2 Minimum Detectable Contrast

The luminance-dependent minimum detectable contrast is proposed in [22]. It is derived from Barten's contrast sensitivity function (CSF) [23]. Contrast sensitivity is defined as the inverse of the modulation threshold of a sinusoidal luminance pattern. The CSF at luminance L and frequency u is modeled in [23] as:

$$S(L,u) = \frac{e^{-2\pi^2\sigma^2u^2}/\kappa}{\sqrt{\frac{2}{T}\left(\frac{1}{X_o^2} + \frac{1}{X_{max}^2} + \frac{u^2}{N_{max}^2}\right)\left(\frac{1}{\eta pE} + \frac{\Phi_0}{1-e^{-(u/u_0)^2}}\right)}}, \tag{2.5}$$

where $\sigma = \sqrt{\sigma_0^2 + (C_{ab}\,d)^2}$ arc min,

$d = 5 - 3\tanh\left(0.4\log(LX_0^2/40^2)\right)$ mm,

$E = \frac{\pi d^2}{4} L\left(1 - (d/9.7)^2 + (d/12.4)^4\right)$ Td,

and where $\kappa = 3$, $\sigma_0 = 0.5$ arc min, $u_0 = 7$ cycles/deg, $C_{ab} = 0.08$ arc min/mm, $X_{max} = 12^\circ$, $T = 0.1$ sec, $N_{max} = 15$ cycles, $\eta = 0.03$, $\Phi_0 = 3 \times 10^{-8}$ sec deg^2, $p = 1.2 \times 10^6$ photons $\cdot$ sec^{-1} $\cdot$ deg^{-2} $\cdot$ Td^{-1}. X_o is usually set to 40°.

To find the minimum detectable contrast at luminance L, the highest sensitivity is

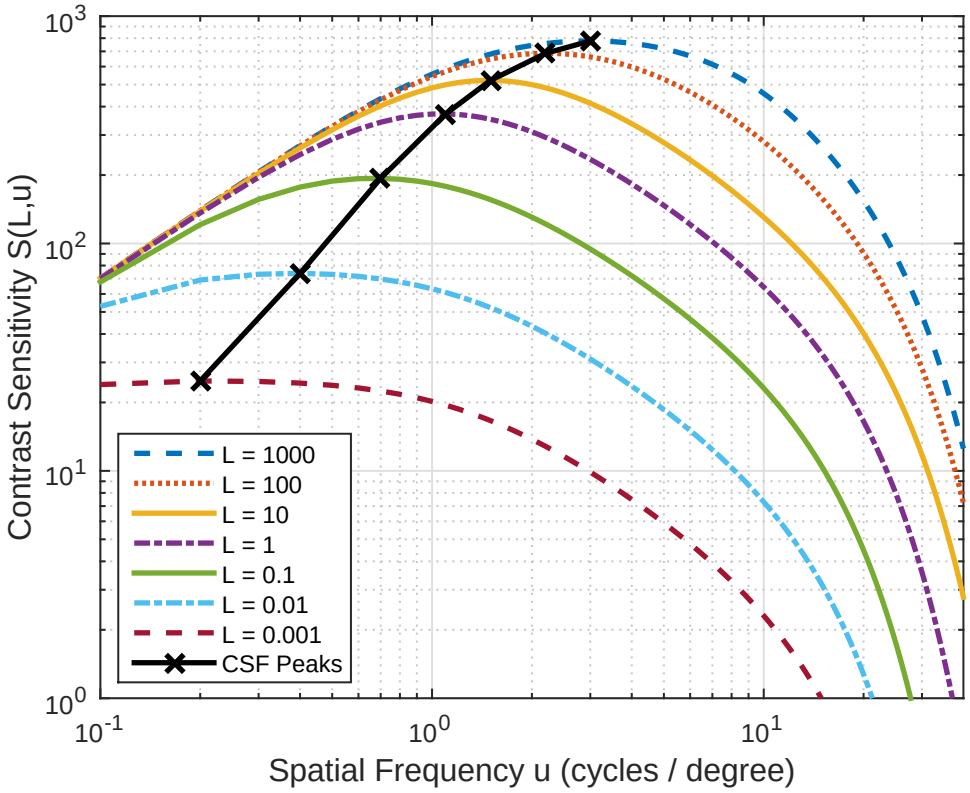


Figure 2.2: Tracking the peaks of contrast sensitivity [7]

found over frequency [7]:

$$S_{max}(L) = \max_{u} S(L,u)\,. \tag{2.6}$$

Fig. 2.2 shows the tracking of peaks of contrast sensitivity when adjusting luminance levels [7]. The minimum detectable contrast $C_t(L)$ for every luminance level is calculated in [7] as:

$$C_t(L) = \frac{1}{S_{max}(L)} \times \frac{2}{1.27}\,, \tag{2.7}$$

where the factor 2 is used for the conversion from modulation to contrast, and the factor $1/1.27$ is used for the conversion from sinusoidal to rectangular waves [22]. Fig. 2.3a shows $C_t(L)$ (solid blue curve) which is called the "Barten ramp" in [22, 7]. Note that the CSF in (2.5) is for the scenario where human eyes are fully adapted to the luminance L.

2.1.3 Ambient-Affected Perceptual and Display Contrast

When the eyes are adapted to some other luminance L_s, the CSF model is modified in [23] as:

$$\tilde{S}(L,u,L_s) = S(L,u)\cdot e^{-\frac{\ln^2\left(\frac{L_s}{L}(1+\frac{144}{x_o^2})^{0.25}\right)-\ln^2\left((1+\frac{144}{x_o^2})^{0.25}\right)}{2\ln^2(32)}}. \tag{2.8}$$

Following the procedure of constructing the minimum detectable contrast for full adaptation, we construct the minimum detectable contrast when the eyes are adapted to L_s. The peaks of contrast sensitivity are found using:

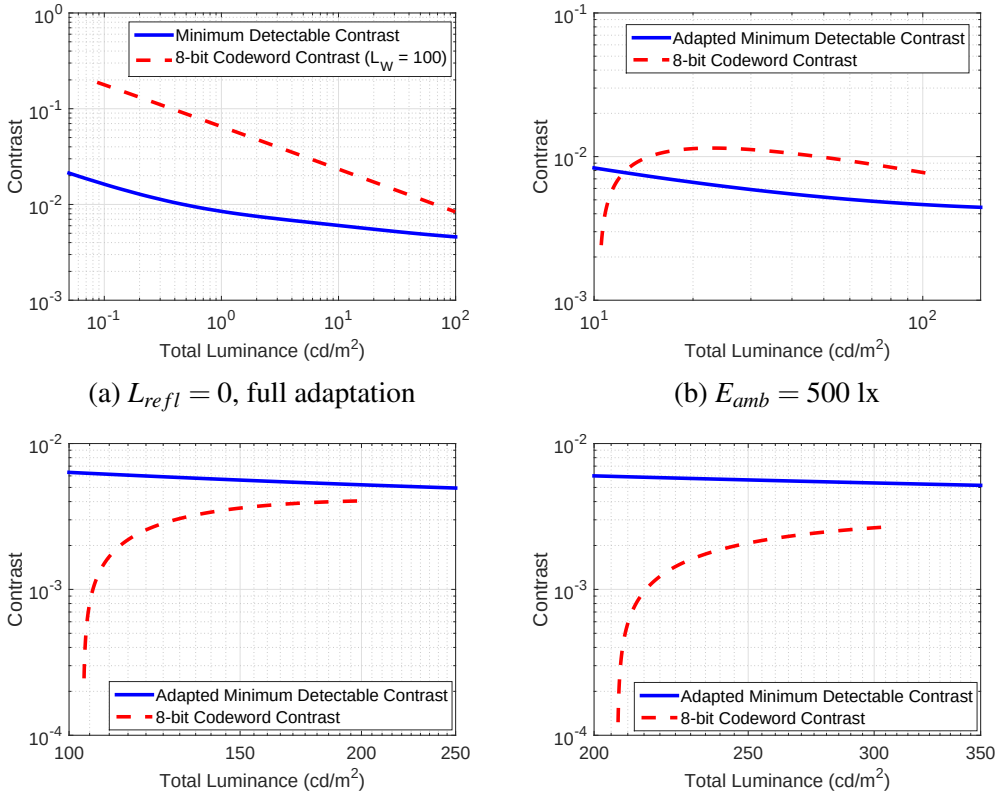
$$\tilde{S}_{max}(L,L_s) = \max_u \tilde{S}(L,u,L_s)\,. \tag{2.9}$$

The adapted minimum detectable contrast $\tilde{C}_t(L,L_s)$ is computed as:

$$\tilde{C}_t(L,L_s) = \frac{1}{\tilde{S}_{max}(L,L_s)} \times \frac{2}{1.27}\,. \tag{2.10}$$

Fig. 2.3b - 2.3d show the adapted minimum detectable contrast when human eyes are adapted to the ambient illumination 500 lx (bright office), 5000 lx (outdoor in shade) and 10,000 lx (overcast day) where $L_s = \frac{E_{amb}}{\pi}$. The codeword contrast of 8-bit displays under the ambient illumination is also plotted using (2.4). As the ambient illumination gets brighter, the adapted minimum detectable contrast increases, and the codeword contrast decreases, thus the codeword contrast gradually drops below the adapted minimum detectable contrast. Note that the codeword contrast in Fig. 2.3 is plotted over the range of the total luminance, $[L_B(L_W)+\frac{k}{\pi}E_{amb}, L_W+\frac{k}{\pi}E_{amb}]$. Under 5000 lx and 10,000 lx, all the codeword contrasts are below the adapted minimum detectable contrast. A codeword contrast lower than the adapted minimum detectable contrast indicates the difference between a codeword and the next codeword cannot be perceived by human eyes. That

results in the reduction or loss of perception of details in bright ambient light. The contrast of dark codewords drops more significantly, yielding more perception loss.



(a) $L_{refl} = 0$, full adaptation

(b) $E_{amb} = 500$ lx

(c) $E_{amb} = 5000$ lx

(d) $E_{amb} = 10,000$ lx

Figure 2.3: Display codeword contrast and minimum detectable contrast of the ideal situation (no reflected light, full adaptation to each luminance) and adaptive minimum detectable contrast under ambient illumination 500 lx, 5000 lx and 10,000 lx for $L_W = 100\,\mathrm{cd/m^2}$.

2.2 Proposed Luminance Enhancement

We want to enhance videos under bright ambient light so that more details become visible. The EOTF of the display is fixed, and the screen luminance for white (L_W) is pre-determined. Our goal is to find a tone mapping function to improve the contrast.

From the last section, it is known that the contrast of codewords decreases,

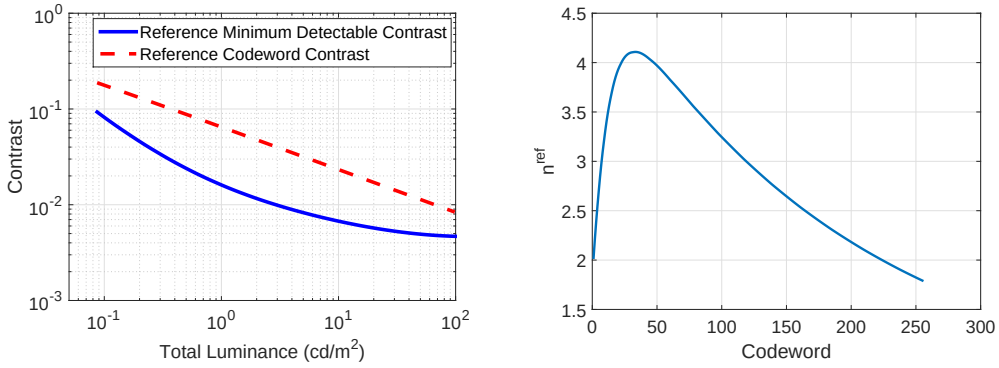
especially for the dark codewords, when the ambient light gets brighter. Note that the contrast ratio of a display is $\frac{L_{total}(255,L_W,E_{amb})}{L_{total}(0,L_W,E_{amb})}$, which is determined by L_W and E_{amb}. For a given L_W under a given E_{amb}, it is not possible to enhance the contrast of every codeword. The contrast of dark codewords is reduced more than bright codewords under bright ambient illumination, so we will enhance the contrast of dark codewords. That means the contrast of some other codewords will be sacrificed.

The tone mapping function will be built so that the tone mapped codewords would have similar contrast distribution to that under the reference viewing condition. The following is defined as the reference viewing condition: the screen brightness for white (L_W^{ref}) is 100 cd/m^2 which is typical for 8-bit displays; the ambient is dark; and eyes are adapted to $L_s^{ref} = (L_W^{ref} + L_B(L_W^{ref}))/2 = (100 + L_B(100))/2$.

Using these settings, we compute the adapted minimum detectable contrast using (2.10) and the codeword contrast using (2.4), and plot them in Fig. 2.4a. We define *relative codeword contrast* as the codeword contrast in the unit of the adapted minimum detectable contrast, which represents the number of just-noticeable-differences (JNDs) that the difference between each two consecutive codewords spans. We define the *reference relative codeword contrast* for each codeword as the relative codeword contrast under the reference viewing condition:

$$n^{ref}(Y,L_W^{ref}) = \frac{C_d(Y,L_W^{ref},0)}{\tilde{C}_t\left(L_{total}(Y,L_W^{ref},0),L_s^{ref}\right)}, \tag{2.11}$$

where $Y = 0,1,\cdots,254$. The relative codeword contrast represents the number of just-noticeable-differences (JNDs) that the difference between each two consecutive codewords spans. The reference relative codeword contrast is plotted in Fig. 2.4b.



(a) Reference contrast (b) n^{ref}

Figure 2.4: Reference contrast and reference relative codeword contrast

2.2.1 Content Independent Luminance Enhancement

A content independent tone mapping operator is proposed in this section. It does not use any video-related data, but only uses display characteristics (e.g., luminance for white L_W, gamma γ and reflectivity k) and the ambient illumination E_{amb}. For a given display and a given ambient illumination, this tone mapping operator can be globally applied to any video or image. Very light computation is needed.

Using the reference relative codeword contrast n^{ref} under the reference condition, we propose to allocate codewords so that the relative contrast at each codeword would be equal to αn^{ref}, where α is positive. That is, the relative codeword contrast should satisfy:

$$\frac{C_d(T^G(Y),L_W,E_{amb})}{\tilde{C}_t\left(L_{total}\left(T^G(Y),L_W,E_{amb}\right),\frac{E_{amb}}{\pi}\right)}=\alpha n^{ref}(Y,L_W^{ref}), \tag{2.12}$$

where $T^G(Y)$ is the output of tone mapping, i.e., $T^G(\cdot)$ is the (global) content independent tone mapping operator. Combining (2.4) and (2.12), we obtain:

$$\frac{2\frac{L_{total}\left(T^G(Y+1),L_W,E_{amb}\right)-L_{total}\left(T^G(Y),L_W,E_{amb}\right)}{L_{total}\left(T^G(Y+1),L_W,E_{amb}\right)+L_{total}\left(T^G(Y),L_W,E_{amb}\right)}}{\tilde{C}_t\left(L_{total}\left(T^G(Y),L_W,E_{amb}\right),\frac{E_{amb}}{\pi}\right)}=\alpha n^{ref}(Y,L_W^{ref}), \tag{2.13}$$

The total luminance of $T^G(Y+1)$ is obtained as a function of the total luminance of $T^G(Y)$:

$$L_{total}(T^G(Y+1),L_W,E_{amb}) = L_{total}\big(T^G(Y),L_W,E_{amb}\big)\frac{2+P(Y,L_W,E_{amb},\alpha)}{2-P(Y,L_W,E_{amb},\alpha)},$$
$$\text{where } P(Y,L_W,E_{amb},\alpha) = \alpha n^{ref}(Y,L_W^{ref})\tilde{C}_t\Big(L_{total}\big(T^G(Y),L_W,E_{amb}\big),\frac{E_{amb}}{\pi}\Big). \tag{2.14}$$

Therefore, the total luminance of each codeword can be derived recursively from that of the previous codeword. The inverse of the display luminance model (2.3) is then applied to compute the tone mapping operator:

$$T^G(Y) = \Big(\big(\frac{L_{total}(T^G(Y),L_W,E_{amb}) - \frac{k}{\pi}E_{amb}}{a(L_W)}\big)^{\frac{1}{\gamma}} - b(L_W)\Big)\cdot 255. \tag{2.15}$$

In most videos, a large portion of pixels are in the mid-tone. Therefore, very dark and very bright codewords can be clipped to further improve the contrast of mid-tones. That is, we set the total luminance of the codewords below z ($0 \le z < 255$) to the total luminance of black ($L_{total}(0,L_W,E_{amb})$). The total luminance of codewords in $[z,255-z]$ is derived recursively from the previous codeword. The total luminance of the codewords above $255-z$ is set to $L_{total}(T^G(255-z),L_W,E_{amb})$. In other words, a number of z codewords at both ends are clipped. In summary, the total luminance of codewords are computed as:

$$L_{total}(T^G(Y),L_W,E_{amb})$$
$$= \begin{cases} L_{total}(0,L_W,E_{amb}) & \text{if } Y < z \\ L_{total}\big(T^G(Y-1),L_W,E_{amb}\big)\frac{2+P(Y-1,L_W,E_{amb},\alpha)}{2-P(Y-1,L_W,E_{amb},\alpha)} & \text{else if } z \le Y \le 255-z \\ L_{total}(T^G(255-z),L_W,E_{amb}) & \text{otherwise} \end{cases}, \tag{2.16}$$

Now the problem is to find α. We want to enhance the relative codeword contrast as much as possible, i.e., we want α to be as large as possible. Note that if $L_{total}(T^G(255-z), L_W, E_{amb})$ exceeds the selected luminance for white (L_W), more codewords would be clipped. In order to avoid that, we formulate the problem as:

$$\begin{aligned} &\max \quad \alpha \\ &\text{s.t.} \;\; L_{total}(T^G(255-z), L_W, E_{amb}) \leq L_W \end{aligned} \tag{2.17}$$

The problem can be solved easily by bisection search. If the contrast ratio of the actual viewing condition is lower than the contrast ratio of the reference condition, i.e.,

$$\frac{L_{total}(255, L_W, E_{amb})}{L_{total}(0, L_W, E_{amb})} < \frac{L_{total}(255, 100, 0)}{L_{total}(0, 100, 0)}, \tag{2.18}$$

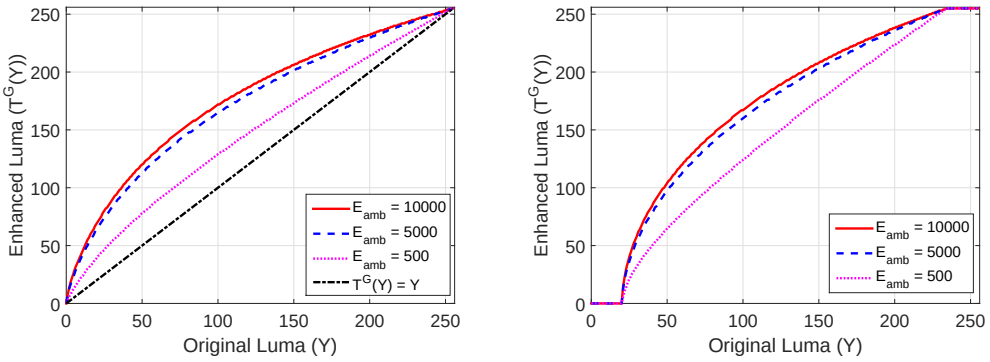
then the optimum α would be less than 1. Fig. 2.5 shows the tone mapping curves for $L_W = 200$ and $E_{amb} = 500$, 5000 and $10,000$, when z is 0 and 20. $T^G(Y) = Y$ (linear mapping) is also plotted as a reference. Compared to the codeword contrast before tone mapping, the contrast of dark codewords after tone mapping is enhanced, whereas the contrast of bright codewords is suppressed. In other words, the contrasts of codewords are re-allocated by the tone mapping. The contrast of dark codewords is more enhanced when the ambient illumination is higher.

Bright ambient light reduces the saturation of the video as well as the luminance contrast. We enhance the chrominance of the video by applying the simple method from [15]:

$$R(V) = \left(\frac{V}{Y}\right)^s T^G(Y), \tag{2.19}$$

where V is the chroma value, $R(V)$ is the enhanced chroma value, and s is a constant.

Fig. 2.6 shows one frame before and after tone mapping using the curve in Fig. 2.5b and (2.19) where $s = 0.8$. The enhanced images look "brighter" and more



(a) $z = 0$ (b) $z = 20$

Figure 2.5: Tone mapping curves of proposed content independent luminance enhancement method for $L_W = 200$: $T^G(Y)$ vs. Y

saturated than the original image under the given ambient illumination, and more details in dark areas are revealed.

2.2.2 Content Dependent Luminance Enhancement

In this section, we describe a content dependent tone mapping operator. In addition to the display characteristics and the ambient illumination, some statistics of the video are collected to construct the tone mapping operator, so that the video can be enhanced better. Unlike the method of Mantiuk et al. [15] where the Laplacian pyramid is employed and the contrast probabilities are computed for every luminance range and every frequency, we simply collect the histograms of codewords of the video.

For a frame, the histograms of codewords are collected as:

$$\begin{aligned} h_{f,m} &= |\Phi_{f,m}|, \\ \text{where} \quad \Phi_{f,m} &= \{j \,|\, \frac{256m}{M} \leq I_{f,j} < \frac{256(m+1)}{M}\}, \end{aligned} \tag{2.20}$$

and where $I_{f,j}$ is the j-th pixel in frame f, and $m = 0, 1, \cdots, M-1$. We set M to 32, i.e., there are 32 bins in the histograms.

(a) Original (b) Enhanced: $E_{amb} = 500$

(c) Enhanced: $E_{amb} = 5000$ (d) Enhanced: $E_{amb} = 10,000$

Figure 2.6: Images before and after the proposed tone mapping for $L_W = 200$. Note that these are inputs to the device, which do not show the look under the given ambient illumination.

We average the histograms of frames from f_x to f_y, and compute the weighting factor for codeword Y as:

$$w(Y) = \left(\frac{1}{f_y - f_x + 1} \sum_{i=f_x}^{f_y} \left(\frac{h_{i,b}}{\sum_{m=0}^{M-1} h_{i,m}} \right) \right)^{\beta}, \quad \text{where} \quad b = \left\lfloor \frac{Y}{256} M \right\rfloor, \tag{2.21}$$

where β is a constant ($0 < \beta \leq 1$), so $w(Y)$ is in $[0,1]$. The frames from f_x to f_y can correspond to a scene or a sliding window which includes the frame f. For example, f_x can be $f-9$ and f_y can be f, and thus the histograms of 10 frames are averaged to give the weighting factors.

The tone mapping operator is constructed using the weighted reference relative

codeword contrast:

$$\frac{C_d(T^D(Y),L_W,E_{amb})}{\tilde{C}_t\left(L_{total}\left(T^D(Y),L_W,E_{amb}\right),\frac{E_{amb}}{\pi}\right)}=\alpha w(Y)\cdot n^{ref}(Y,L_W^{ref}), \tag{2.22}$$

where $T^D(\cdot)$ is the tone mapping operator. Compared to (2.12), the codewords are allocated so that the relative codeword contrast satisfies $\alpha w(Y)\cdot n^{ref}(Y,L_W^{ref})$ instead of $\alpha n^{ref}(Y,L_W^{ref})$. The contrast of the codewords corresponding to higher histogram counts is enhanced more, because these codewords take up larger areas in the video. For example, say the codewords under 50 take up 80% of a video. They are likely to be more perceptually important than the other codewords. The contrast of the codewords under 50 are more enhanced by multiplying the reference relative codeword contrast by larger weighting factors.

Combining (2.4) and (2.22), the contrast between two consecutive codewords after tone mapping is hence constructed as:

$$\begin{aligned}2\frac{L_{total}\left(T^D(Y+1),L_W,E_{amb}\right)-L_{total}\left(T^D(Y),L_W,E_{amb}\right)}{L_{total}\left(T^D(Y+1),L_W,E_{amb}\right)+L_{total}\left(T^D(Y),L_W,E_{amb}\right)}&\\ =\alpha w(Y)\cdot n^{ref}(Y,L_W^{ref})\cdot\tilde{C}_t\left(L_{total}\left(T^D(Y),L_W,E_{amb}\right),\frac{E_{amb}}{\pi}\right),&\end{aligned} \tag{2.23}$$

We obtain the total luminance of $T^D(Y+1)$ as:

$$\begin{aligned}&L_{total}(T^D(Y+1),L_W,E_{amb})=L_{total}\left(T^D(Y),L_W,E_{amb}\right)\frac{2+Q(Y,L_W,E_{amb},\alpha)}{2-Q(Y,L_W,E_{amb},\alpha)},\\ &\text{where }\ Q(Y,L_W,E_{amb},\alpha)=\alpha w(Y)\,n^{ref}(Y,L_W^{ref})\tilde{C}_t\left(L_{total}\left(T^D(Y),L_W,E_{amb}\right),\frac{E_{amb}}{\pi}\right).\end{aligned} \tag{2.24}$$

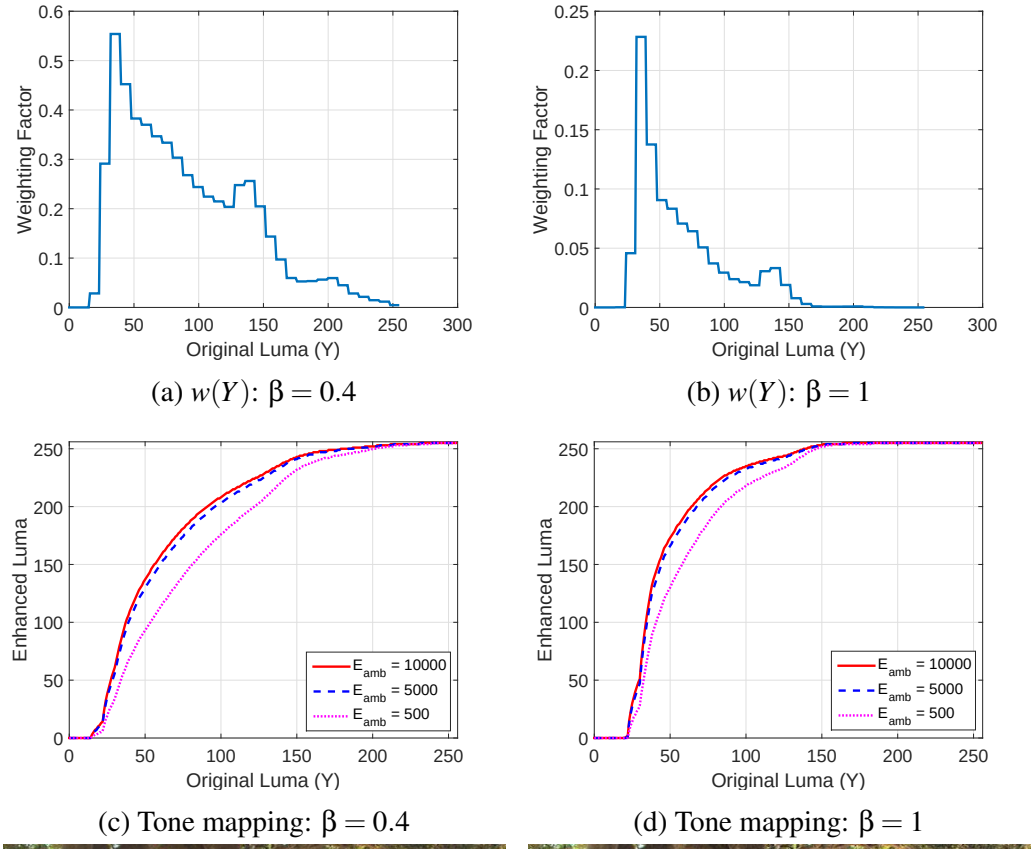
$T(0)$ is set to 0, i.e., the total luminance of the first codeword, $L_{total}(T^D(0),L_W,E_{amb})$,

is set to $L_{total}(0, L_W, E_{amb}) = L_B(L_W) + \frac{k}{\pi} E_{amb}$. The total luminance of other codewords is derived recursively from that of the previous codeword. The problem is formulated similarly to (2.17):

$$\begin{aligned} &\max \quad \alpha \\ &\text{s.t.} \quad L_{total}(T^D(255), L_W, E_{amb}) \leq L_W \end{aligned} \tag{2.25}$$

The tone mapping operator, $T^D(Y)$, is then obtained by applying (2.15) where $T^G(Y)$ is replaced by $T^D(Y)$. The chrominance of the video is enhanced similarly to (2.19): $R(V) = \left(\frac{V}{Y}\right)^s T^D(Y)$.

Fig. 2.7 shows the weighting factors and the tone mapping curves for Fig. 2.6a when $\beta = 0.4$ and $\beta = 1$. L_W is 200 cd/m^2. Figs. 2.7e and 2.7f show the corresponding enhanced images when $E_{amb} = 5000$. The tone mapping curve of $\beta = 1$ is almost flat for the codewords above 150, thus yielding detail loss in bright areas in Fig. 2.7f (the texture on the ground is removed). The reason is that the weighting factors of those bright codewords are much smaller than the weighting factors of dark codewords, and thus the contrast of the bright codewords is ignored in the optimization. When β is 0.4, the variance of the weighting factors is reduced, and thus the bright codewords have more impact on the tone mapping curve. As a result, the details in the bright regions are preserved better. Note that the codewords which correspond to short bins in the histograms should not be ignored completely, because they can be the foreground in the picture thus drawing the viewer's attention. Compared to the result of content independent luminance enhancement (Fig. 2.6c), Fig. 2.7e looks brighter, though it requires a bit higher computation to collect the histograms. Unlike the proposed content independent method, here codewords at both ends are clipped according to the histograms rather than selected heuristically.



(a) $w(Y)$: $\beta = 0.4$

(b) $w(Y)$: $\beta = 1$

(c) Tone mapping: $\beta = 0.4$

(d) Tone mapping: $\beta = 1$

(e) Enhanced image: $\beta = 0.4$

(f) Enhanced image: $\beta = 1$

Figure 2.7: Weighting factors, tone mapping curves, and enhanced images for $E_{amb} = 5000$ when β is 0.4 and 1

2.3 Performance Evaluation

We evaluate the performance of the tone mapping operators by a subjective test. Six video clips [24, 25] are enhanced using our proposed methods, the tone mapping method of Mantiuk et al. [15], and the adaptive luminance enhancement method of Su et

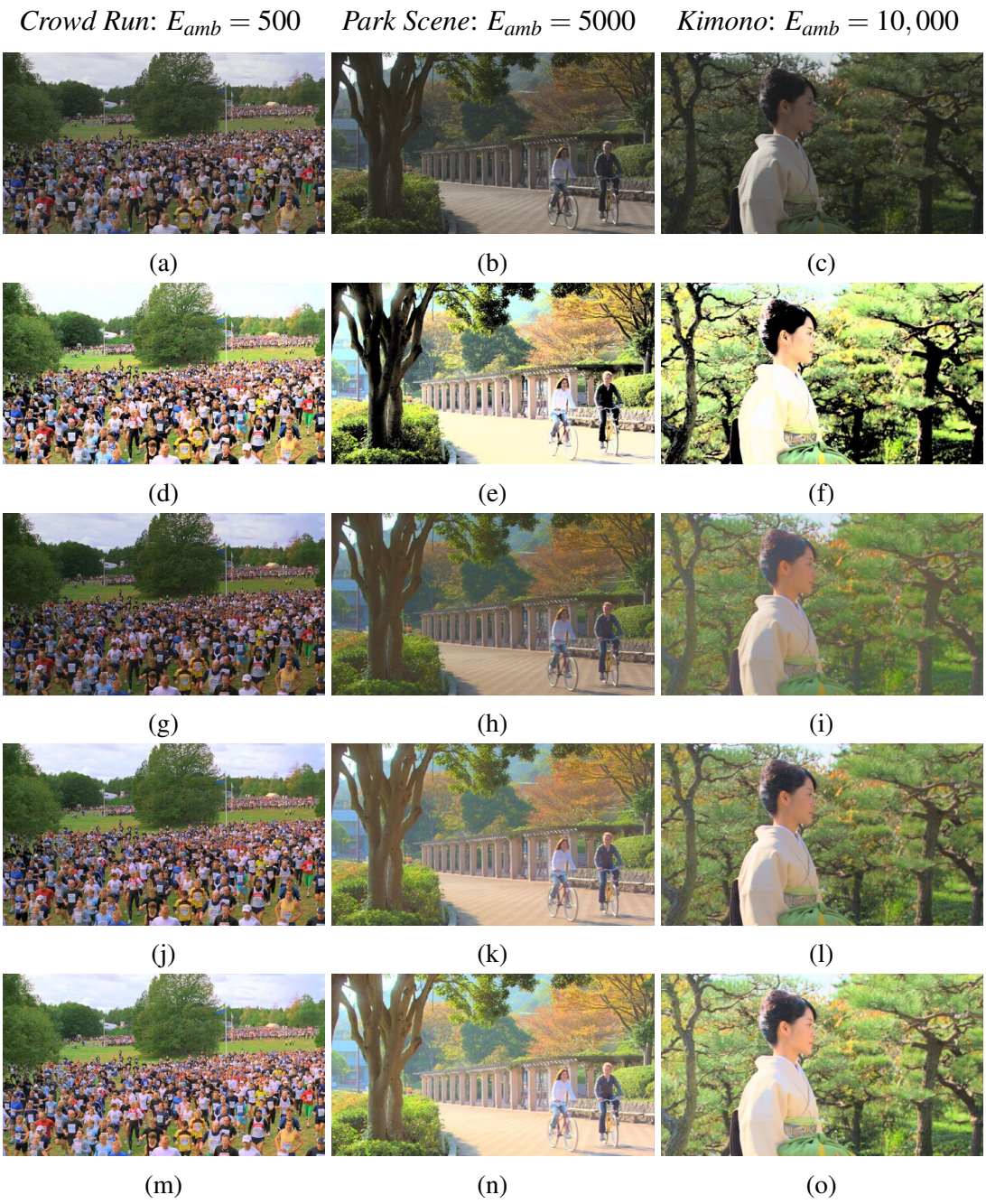


Figure 2.8: Images before and after tone mapping using different algorithms for $L_W = 200$. (a) - (c): original; (d) - (f): Mantiuk et al. [15]; (g) - (i): Su et al. [20]; (j) - (l): proposed content independent enhancement; (m) - (o): proposed content dependent enhancement. Note that these are the inputs to the device, which do not show the look under the given ambient illumination.

al. [20]. Each video has only one scene, and the content does not change dramatically. Our proposed method in Sec. 2.2.1 and the method of Su et al. are content independent, i.e., the tone mapping is global for all videos, so they do not have any temporal issue. The method of Mantiuk et al. is a frame-wise solution, which may cause temporal flickering due to differences in tone mapping of frames. The weighting factors of the proposed content dependent method in Sec. 2.2.2 are computed using the average histograms of all the frames of the video, and one tone mapping operator is built for the video, so the tone mapping does not cause any temporal issue. If the sliding window in (2.21) is too small (e.g., only one frame), there can be temporal flickering. The temporal flickering is not the interest of this work. In order to reduce the length of experiments and tiredness of subjects, we had subjects watch images extracted from videos, instead of watching the whole videos. One image is extracted from each video.

The main focus of this work is on luminance enhancement, not on chrominance adjustment. To rule out the effects of different chrominance adjustment methods, chroma is enhanced using the same method in (2.19) for all the tone mapping methods, where $T^G(Y)$ is replaced by each tone mapping operator. Some enhanced images are shown in Fig. 2.8.

Pair comparison [11, 10] is used to evaluate the performance of methods. Subjects compared images in pairs: one image processed by one of our proposed methods, and the other by one of the baseline schemes which include the method of Mantiuk et al., the method of Su et al., and no processing (the original image). The two proposed methods are also compared with each other. The images were labeled A and B randomly. Subjects were given 5 options: “A is much better than B”, “A is slightly better than B”, “A is the same as B”, “A is slightly worse than B”, and “A is much worse than B”. Subjects were also asked to select the reasons why they prefer one to the other one. The possible reasons were: being brighter, being darker, more details, higher contrast, or lower contrast.

The images were shown on an iPad Air. The reflectivity is 6.5%, the display gamma is 2.23, and the screen brightness for white is adjustable from 6 to 449 $\mathrm{cd/m^2}$ [14]. The whole experiment took about 30 minutes, which included four sessions of viewing conditions:

1) L_W is 200 $\mathrm{cd/m^2}$ and E_{amb} is 500 lx,

2) L_W is 200 $\mathrm{cd/m^2}$ and E_{amb} is 5000 lx,

3) L_W is 200 $\mathrm{cd/m^2}$ and E_{amb} is 10,000 lx,

4) L_W is 449 $\mathrm{cd/m^2}$ and E_{amb} is 10,000 lx.

2.3.1 Results of Content Independent Enhancement

Nine subjects conducted the comparisons between the proposed content independent method and the baseline schemes. When our proposed method was rated much better (or worse) than the other scheme, the opinion score is +2 (or -2); when our proposed method was rated slightly better (or worse) than the other scheme, the opinion score is +1 (or -1); when no difference was found, the opinion score is 0. The difference mean opinion score (DMOS) is computed between the proposed method and the other schemes. Positive (or negative) numbers mean the proposed luminance enhancement works better (or worse) than the other scheme. We plot the DMOS and the 95% confidence intervals (CIs) in Fig. 2.9, where the first row is the DMOS against no processing, the second is against the method of Mantiuk et al., and the third is against the method of Su et al. Each column corresponds to one tested viewing condition. The average DMOS of all the images and CIs are plotted in Fig. 2.10, where "original" means no processing. CIs including zero indicate that we cannot reject the null hypothesis that the two schemes perform the same, or at least there is no consensus of preference.

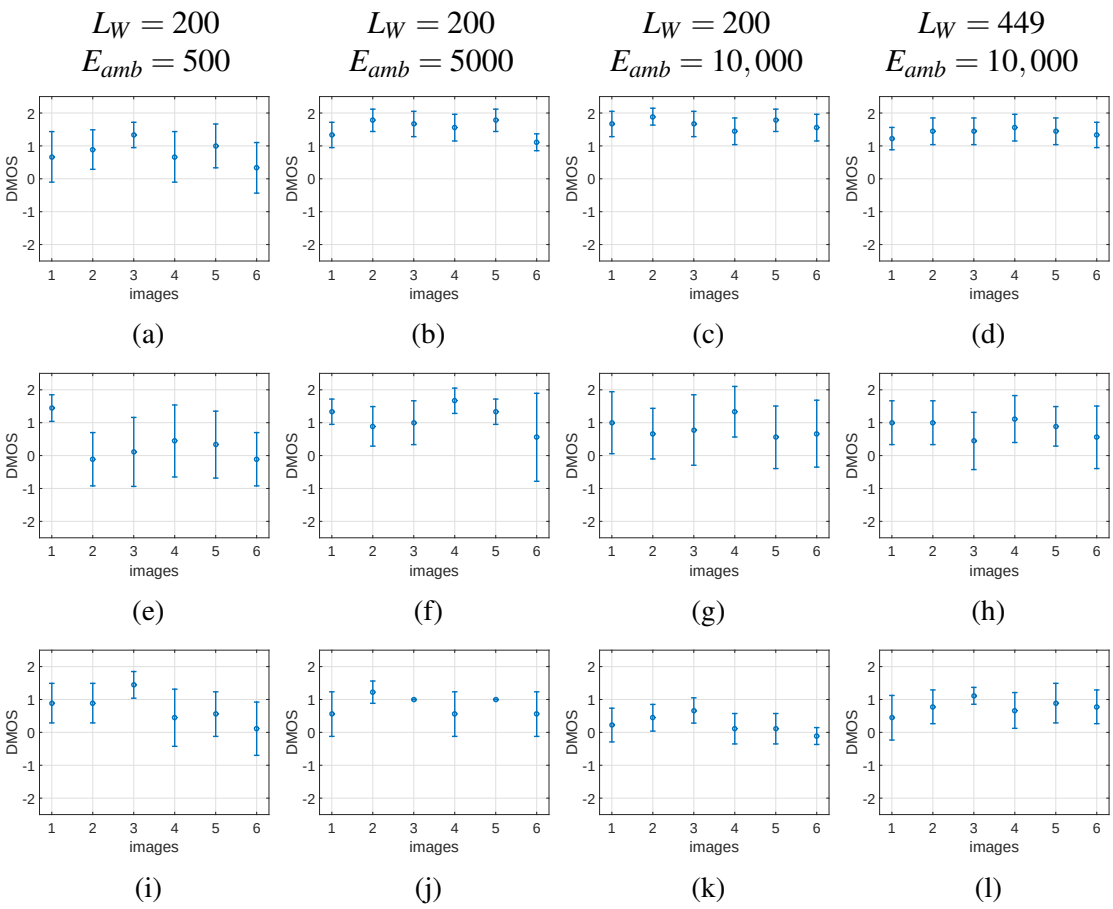
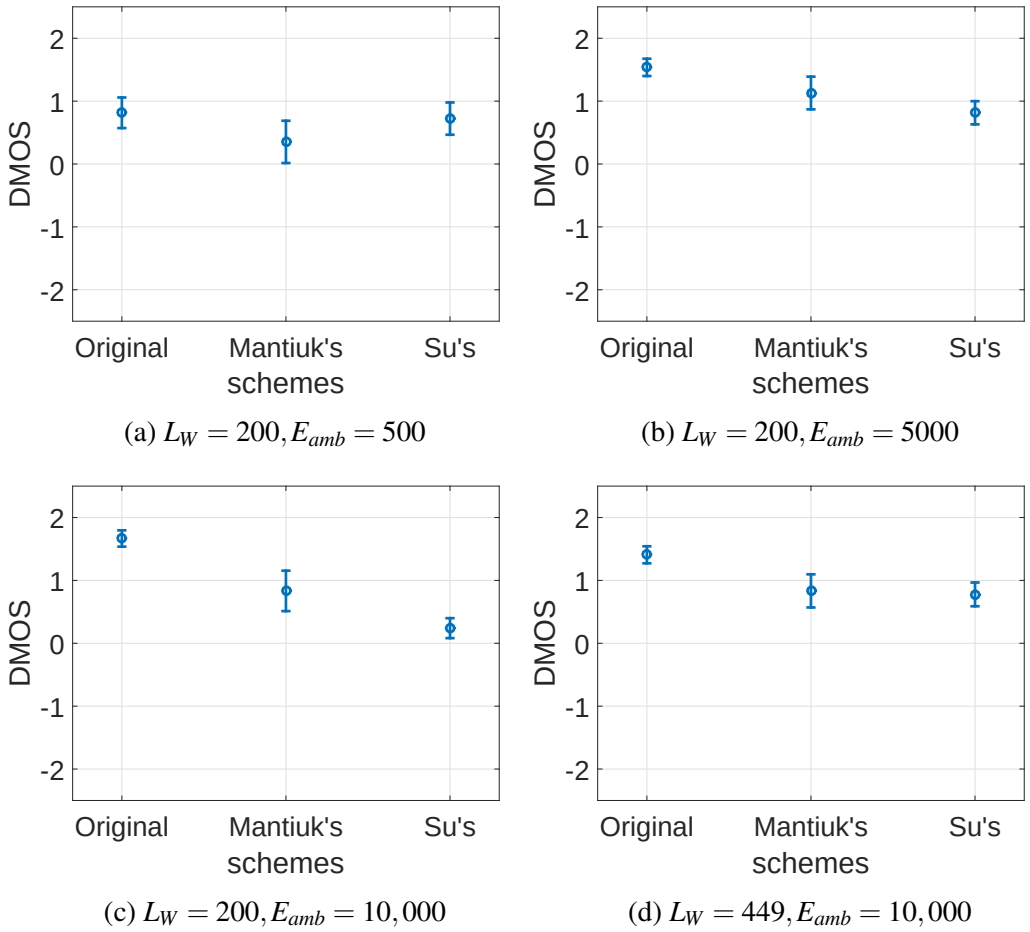


Figure 2.9: 95% confidence intervals of DMOS of proposed content independent luminance enhancement method vs. other schemes. (a) - (d): against no processing (original), (e) - (h): against the method of Mantiuk et al., (i) - (l): against the method of Su et al. Images 1 - 6 are *Crowd Run*, *Into Tree*, *Kimono*, *Old Town Cross*, *Park Scene* and *Rush Hour*.

When L_W is 200 cd/m^2, the enhanced images of the proposed content independent method have higher gain over the original images as the ambient light gets brighter. Under 500 lx (Fig. 2.9a), the CIs of 3 out of 6 images are above zero; while under 5000 lx and 10,000 lx (Figs. 2.9b and 2.9c), the CIs of all the 6 images are above zero. The average DMOS over images are 0.81, 1.54 and 1.67 for 500 lx, 5000 lx and 10,000 lx, respectively. Even when L_W is set to the maximum screen brightness 449 cd/m^2, the proposed scheme outperforms no processing by an average DMOS of 1.41, under 10,000 lx. The original images look dark and dull with low contrast under bright ambient light,



(a) $L_W = 200, E_{amb} = 500$ (b) $L_W = 200, E_{amb} = 5000$

(c) $L_W = 200, E_{amb} = 10,000$ (d) $L_W = 449, E_{amb} = 10,000$

Figure 2.10: 95% confidence intervals of average DMOS of proposed content independent luminance enhancement method vs. other schemes for all images.

and details are invisible. The proposed method improves the visibility, and enhances the brightness of images.

The method of Mantiuk et al. generally boosts the contrast higher than our proposed method, which yields brighter images and sharper edges, but sometimes removes details in bright areas of images. For example, the clouds in Fig. 2.8d, the details of the ground in Fig. 2.8e, and the clothing shades in Fig. 2.8f are removed. More details are lost when the ambient light is brighter. Our proposed method (Figs. 2.8j - 2.8l) preserves those details very well.

The preference between the method of Mantiuk et al. and our proposed content

independent method is controversial under 500 lx. The CIs are wide and include zero for 5 out of 6 images in Fig. 2.9e. Subjects have various preferences in contrast and details. The proposed method shows clear advantage for image *Crowd Run*, as most subjects valued the details preserved by the proposed method.

The proposed method is favored over the method of Mantiuk et al. for most images under 5000 lx, due to more details and lower contrast. The advantage drops when the ambient increases to 10,000 lx with L_W kept to 200 $\mathrm{cd/m^2}$, because the favor of subjects shifts toward high contrast under such bright ambient where details in most images are hardly detectable. However, when L_W is turned up to 449 $\mathrm{cd/m^2}$, our method which has higher visibility of details wins the comparison. When we pool the DMOS of all the test images (Fig. 2.10), the proposed method outperforms the method of Mantiuk et al. in all the viewing conditions, among which the DMOS of 5000 lx is the highest.

The results of the method of Su et al. are generally darker than the proposed content independent method, and show lower contrast. Under 500 lx and 5000 lx where the eyes of subjects are relatively sensitive, the proposed method outperforms the method of Su et al. for half of the images. For example, the texture of the trees in Fig. 2.8g is undetectable and looks flat, whereas the result of our proposed method (Fig. 2.8j) shows the texture clearly. As a result, the DMOS between our proposed method and the method of Su et al. for this image is 0.99 under 500 lx, and the CI is well above zero (Fig. 2.9i). For another instance, Fig. 2.8h looks washed out under 5000 lx. Details in dark areas are still invisible in the bright surrounding as in the original image Fig. 2.8b.

Under 10,000 lx, the two methods perform similarly for 4 out of 6 images when L_W is 200 $\mathrm{cd/m^2}$, but the proposed method shows clear advantage when L_W is 449 $\mathrm{cd/m^2}$. When we pool all images in Fig. 2.10, the CIs of the average DMOS of all the viewing conditions are above zero, indicating the superiority of our method.

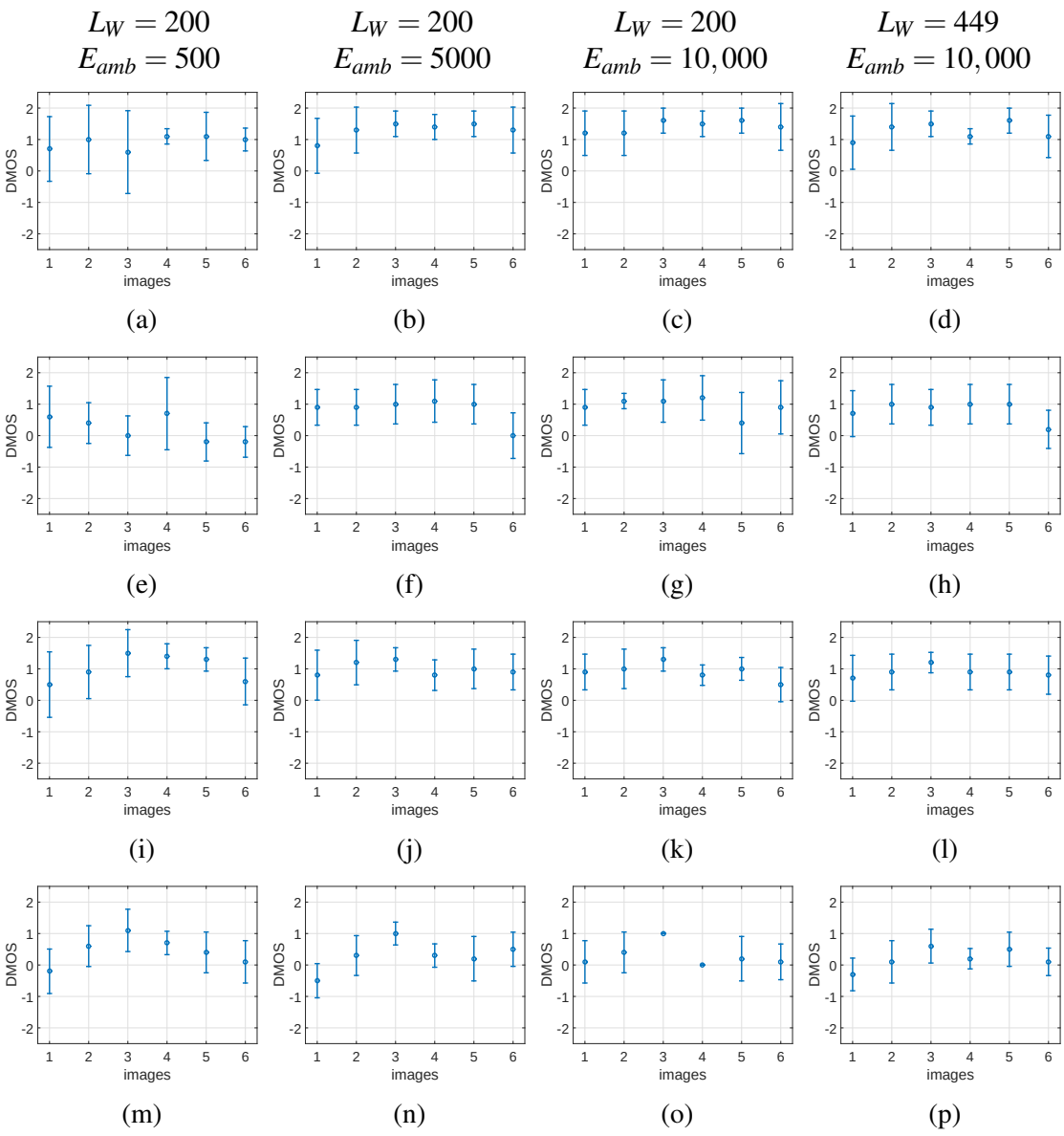
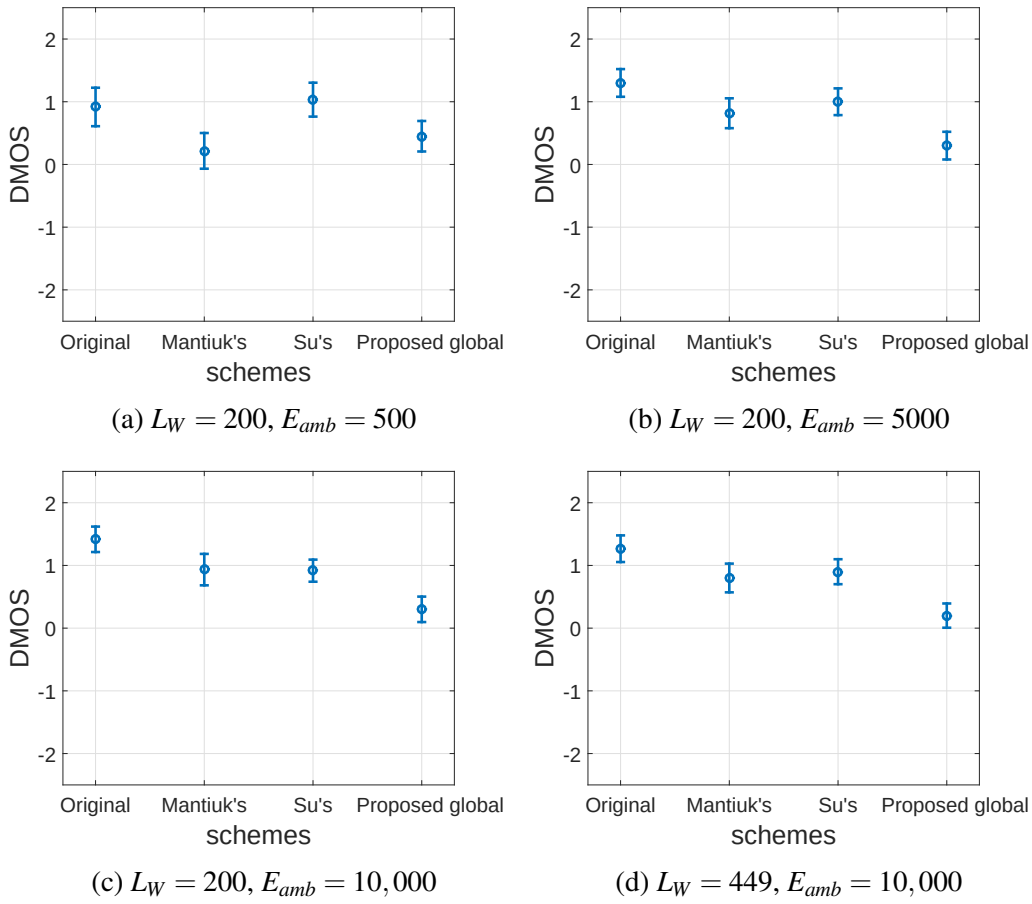


Figure 2.11: 95% confidence intervals of DMOS of proposed content dependent luminance enhancement method vs. other schemes. (a) - (d): against no processing (original), (e) - (h): against the method of Mantiuk et al., (i) - (l): against the method of Su et al., (m) - (p): against the proposed content independent method. Images 1 - 6 are *Crowd Run*, *Into Tree*, *Kimono*, *Old Town Cross*, *Park Scene* and *Rush Hour*.

2.3.2 Results of Content Dependent Enhancement

The proposed content dependent method and the other schemes were evaluated by another nine subjects. The DMOS and CIs of each image are plotted in Fig. 2.11.



(a) $L_W = 200$, $E_{amb} = 500$ (b) $L_W = 200$, $E_{amb} = 5000$

(c) $L_W = 200$, $E_{amb} = 10,000$ (d) $L_W = 449$, $E_{amb} = 10,000$

Figure 2.12: 95% confidence intervals of average DMOS of proposed content dependent luminance enhancement method vs. other schemes for all test images. "Proposed global" means the proposed content independent method.

Each row corresponds to the DMOS against no processing, the method of Mantiuk et al., the method of Su et al., and the proposed content independent method. The images enhanced by the proposed content dependent method generally looks brighter and shows higher contrast than those enhanced by the proposed content independent method. That is because the content dependent method puts more emphasis on the codewords which take up larger areas of the picture. The contrasts of those codewords are enhanced more than the other codewords. The difference in the image *Kimono* (Figs.2.8l and 2.8o) is the most obvious, and thus the DMOS of that image is the highest among all the test images under all the viewing conditions. The CIs of the average DMOS for all the images against

the proposed content independent scheme are all above zero (Fig. 2.12 where "Proposed global" means the proposed content independent method).

Like the proposed content independent method, the content dependent method greatly outperforms no processing, and the advantage grows with the increase of the ambient illumination. Under 500 lx, the proposed content dependent method is preferred for half of the test images; under 5000 lx, it is favored for 5 out of 6 images; and under 10,000 lx, it beats no processing for all the images.

Under 500 lx, the proposed content dependent method performs similarly to the method of Mantiuk et al. The proposed method wins when the ambient light is brighter, because it shows a better trade-off between detail preservation and contrast enhancement. The CIs of 5 out of 6 images are above zero when the ambient illumination is 5000 lx and 10,000 lx.

The subjects preferred the proposed content dependent method to the method of Su et al. for most of the test images. The superiority is quite clear. The average DMOS of the four viewing conditions are 1.03, 1.0, 0.92 and 0.90.

In summary, both of the proposed methods outperform the baseline schemes. They keep details of images better than the method of Mantiuk et al., and they improve the contrast and brightness more than the method of Su et al. Therefore, they win the comparisons when the ambient light is very bright. The advantages of both proposed methods over no processing are quite large, as the proposed methods enhance the brightness and contrast. The proposed content dependent method is slightly better than the content independent method, because the contrast of codewords with higher histogram counts is boosted higher.

Note that both of our proposed methods, content independent and dependent enhancement, are computationally much simpler than the method of Mantiuk et al. which is a content dependent approach. As stated in [15], half of the processing time

of the method of Mantiuk et al. is spent on computing the contrast probabilities of each luminance range of each frequency of the Laplacian pyramid. The other half is spent on solving their optimization problem iteratively. Our content dependent method only collects histograms of codewords and does not do any frequency decomposition. The optimization time is also much shorter than the method of Mantiuk et al. The proposed content independent method is computationally even more efficient, as no video-related data is needed.

2.4 Summary

In this chapter, we propose two tone mapping operators to improve the perceptual quality of videos shown in bright ambient light. The main contributions include:

1. We analyze the effects of ambient light reflection and the reduction of human visual sensitivity on the perceptual quality of videos displayed in bright ambient illumination.

2. A content independent tone mapping operator is constructed by considering reflection and human visual sensitivity. For a given device under a given ambient illumination, the tone mapping operator can be applied to any videos. The computation is very light.

3. A content dependent tone mapping operator is built by using simple statistics of a video in addition to the display characteristics and visual sensitivity. It requires a bit more computation than the content independent method, and outperforms the latter slightly.

4. We conducted subjective tests and compared our proposed methods with the method of Mantiuk et al. [15] and the method of Su et al. [20]. The results show

our methods are preferred, as the details in dark areas of videos are boosted, and details in bright areas are well preserved.

Acknowledgment

The authors would like to thank Dr. Louis J. Kerofsky who helped with display measurement.

Chapter 2, in part, is a reprint of material as it appears in Q. Song and P. C. Cosman, “Luminance and detail enhancement of videos adapted to ambient illumination”, submitted to *IEEE Transactions on Image Processing*. The dissertation author was the primary author of this paper and the co-author Prof. Cosman supervised the research.

Chapter 3

Subjective Quality of Video Bit Rate Reduction by Distance Adaptation

In the previous chapter, we construct tone mapping operators to enhance the perceptual quality of videos displayed under bright ambient light at the decoder (receiver). In addition to ambient illumination, viewing distance also affects the amount of detail that can be perceived. Increasing the detail visibility can be easily achieved by reducing the viewing distance. But for the viewing distance selected by the viewer, transmission of the unnoticeable details is wasteful. In this chapter, we demonstrate the performance of a distance adaptive delivery system by subjective tests. The viewing distance is detected at the receiver and transmitted back to the server. The server hence selects the corresponding video bitstream adapted to the viewing distance and transmits it to the viewer. We show that adapting to conditions of an individual viewer provides a promising area to reduce bit-rate without sacrificing video quality.

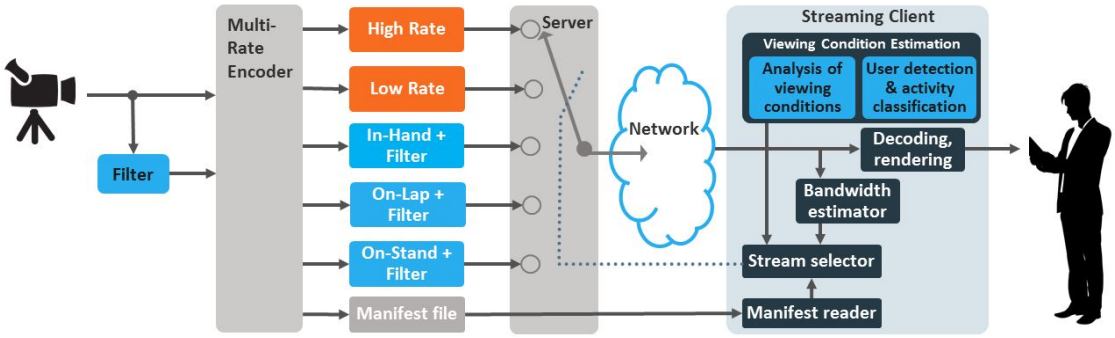


Figure 3.1: Architecture of user-adaptive video delivery system

3.1 Motivation

The same video content may be viewed on any of a variety of devices under dynamically varying viewing conditions. The work of [26] examined typical usage of tablet devices and determined common usage clustered into modes such as On-Lap and On-Stand. These modes correspond to different viewing distances. A user may hold a tablet near while watching a short video clip but discomfort will prevent the user from holding a tablet at a close distance for the duration of long format content.

Compressed bit-rate and video quality are inversely related with the relation depending upon content and viewing conditions. We are interested in exploiting the variation in viewing distance to achieve rate reduction without sacrificing perceived video quality. Xue et al. [27] proposed a strategy to select quantization parameters based on an environment-aware quality assessment model which uses viewing distance, display size, ambient luminance and body movement. Another perceptually motivated technique is to filter the video prior to encoding based on the anticipated viewing conditions. A perceptual pre-filter in [6] removes the spatial oscillations in a video that are invisible under given viewing conditions, resulting in lower complexity images which can be compressed at a lower bit-rate without loss of subjective quality. Bit-rate savings can be easily documented but potential impact on subjective video quality requires visual

testing. That is the goal of this chapter. To evaluate the perceptual quality performance of the pre-filter and the whole user-adaptive video delivery system, we conducted a subjective test based on the pair comparison (stimulus-comparison) method [10, 11]. Observers compared the quality of compressed videos shown on a tablet with and without pre-filtering, and graded each pair's difference. We examined three common viewing distances corresponding to using a tablet on a stand, on the lap, and in the hand.

In Section 3.2 we review the design of a viewing condition adaptive system. In Section 3.3 we describe the subjective testing. Results are in Section 3.4, and Section 3.5 summarizes the chapter.

3.2 Viewer Adaptive System

In conventional video coding and delivery systems, viewing condition parameters are not known and are assumed to be within typical ranges (e.g., viewing distance equal to 3 to 4 times screen height). However, as exemplified in Fig. 3.1, one can design an adaptive system that classifies user state and viewing conditions and then uses them to select one of the available encoded versions of the content (representations) on the HTTP server. The representations may include versions with different pre-filtering applied prior to encoding, as well as traditional encodings performed using different target bit-rates. A special manifest file is also placed on the HTTP server to describe properties of all available representations. In performing stream selection, the client software (media player) may find the best matching encoded video representation given a combination of current viewing conditions and network bandwidth limits. The design of such a user-adaptive video delivery system was first proposed in [28]. The implementation of user-adaptive streaming utilizing an MPEG-DASH streaming standard was described in [29].

As mentioned above, the representations of content may differ in the pre-filtering applied in addition to traditional factors. Given viewing distance, the pre-filter may be used to remove details from the content which would be invisible but still require bits to transmit to the device. The perceptual pre-filter described in [6] exploits three basic phenomena of human vision: (1) Contrast sensitivity function: relationship between frequency and contrast sensitivity thresholds of human vision, (2) Eccentricity: rapid decay of contrast sensitivity as angular distance from gaze point increases, and (3) Oblique effect: lower visual sensitivity to diagonally oriented spatial oscillations as opposed to horizontal and vertical ones.

Fig. 3.2 shows examples of encodings produced with and without perceptual filtering. The encodings in sub-figures (c) and (d) use the same rate, however, the filtered version looks softer with fewer coding artifacts. When viewed from a certain distance, the softness introduced by the pre-filter becomes invisible, but bit-rate savings remain.

3.3 Subjective test

We conducted a subjective test of the performance of the pre-filter using the pair comparison method [10, 11]. HD video source sequences were obtained from [25]. Video clips compressed with and without the pre-filtering are shown sequentially in some randomized order to the subjects who provide a comparative preference score. The videos were displayed on a tablet (Nexus 7). To begin, we defined three viewing modes: In-Hand, On-Lap, and On-Stand. The three viewing modes correspond to three viewing distances, i.e., three sets of filter parameters. For “In-Hand” mode, the device is held in both hands. Subjects sat in an armless chair, so their hands were not steadied against anything. For “On-Lap” mode, the device rests on the lap. Subjects could tilt the device to make a good viewing angle but the device remains on the lap. For “On-Stand” mode,

Figure 3.2: Examples of different encodings (1st frame from Old town sequence): (a) Original uncompressed frame, (b) Compressed at *High* rate, (c) Compressed at *Low* rate, (d) Filtered and Compressed at "On Stand" rate.

the device is on a stand on a table, and the subject does not touch it after the initial comfortable positioning. We assume the viewing distances of In-Hand, On-Lap and On-Stand modes are 12", 20" and 24" respectively [26].

3.3.1 Video Versions

For each viewing mode, we apply the pre-filter to the original uncompressed video. Longer viewing distance results in stronger filtering so that more details are removed. Then the filtered videos are compressed by the x264 encoder [30], configured to produce High-Profile H.264/AVC-compliant bitstreams. We denote the compressed filtered videos as *user adaptive videos (UAV)*.

For comparison, we also compress the original video by the same encoder without pre-filtering. The video is compressed at two bit-rates: one bit-rate (called *High*) is higher

than the highest *UAV* bit-rate, and the other (called *Low*) is approximately equal to the lowest *UAV* bit-rate. *High* and *Low* versions serve as negative and positive controls. The goal is that *UAV* should have quality equivalent to *High*, given the corresponding viewing conditions. However, if only *UAV* and *High* are compared and no difference is found, it is possible that this outcome arose because the observers are sleepy, distracted, or in some way unreliable, or because both data rates are so low (or so absurdly high) that no difference between them can be discerned. So we also compare *Low* with *UAV*, to be able to exclude these possibilities. If the pre-filter works for all modes, the outcome would support that all *UAV* versions have quality equal to that of the unfiltered version *High*, and the *UAV* versions have better quality than the unfiltered version *Low*.

The filtering parameters are based on the viewing modes. The three viewing modes (In-Hand, On-Lap and On-Stand) result in three filtered versions, which are compressed at different bit-rates. Together with the *High* and *Low* bit-rates, each video sequence is compressed at five bit-rates using the following steps:

1. Compress the unfiltered sequence with a high bit-rate such that there is no visual artifact. The full encoding capability of H.264 high profile and 1-pass rate control are used to encode the sequence. The output bitstream is the *High* version.
2. For each viewing mode, compress the filtered sequence with multiple bit-rates. The one that has the lowest bit-rate and is visually very close to *High* under the given viewing conditions is selected. The output bitstreams are the *UAV*: In-Hand, On-Lap, and On-Stand versions.
3. Encode the original unfiltered sequence at a bit-rate which is close to but slightly higher than the rate of On-Stand. It gives the bitstream *Low* version. The encoder settings except for the bit-rate are the same as the settings in step 1 for all versions.

The five bit-rates are selected manually for each sequence by experts. The relationship of the bit-rates of the five test versions are $High > Hand > Lap > Stand \approx$

Table 3.1: Bit-rate of each test sequence. All sequences are at 25fps with the exception of Kimono which is at 24fps. The bit-rate of *High* is in kb/s, while others are represented as the percentage compared to *High*.

Sequence	Bit-rate				
	High	UAV			Low
		Hand	Lap	Stand	
Basketball	4008	98.7%	76.6%	65.5%	66.2%
Into trees	8414	99.0%	72.8%	62.4%	62.5%
Old town	3420	97.1%	67.7%	54.8%	60.1%
Sunflower	2290	88.0%	66.0%	45.1%	45.8%
Pedestrian	13058	99.7%	80.0%	56.5%	58.1%
Station	3494	99.0%	76.2%	64.1%	66.6%
Tractor	6512	97.8%	67.3%	54.7%	55.9%
Rush hour	6689	97.9%	66.3%	52.2%	55.6%
Kimono	4980	99.2%	63.2%	61.9%	65.7%
Average	-	97.4%	70.7%	57.5%	59.6%

Low. The rates of each version of each test sequence are in Table 3.1.

3.3.2 Comparison Method

We used the pair comparison (stimulus-comparison) method [10, 11] to compare video quality. The subject was presented with a series of sequence pairs, each from the same source, but the rate and/or the compression (with or without filtering) are different. Videos were presented sequentially on the same device. The subject provides a score of the second sequence (test) relative to the first one (reference) of -1 = worse, 0 = same, 1 = better. We did not follow the 7-point grading in [10] as the differences were very subtle. For each mode, the three versions (*UAV, High, Low*) were shown as reference/test in pseudo-random fashion. The comparisons of each viewing mode included, in randomized order, *UAV* vs. *High*, *UAV* vs. *Low*, *High* vs. *Low*, and *High* vs. *High*. The first two comparisons are the main purpose of our test. *High* vs. *Low* provides a sanity check of the results. *High* vs. *High* is a null test to check for subject accuracy. Note that the subjective tests in Chapter 2 and 4 do not include such null tests, because relatively larger

differences are compared in those experiments so the responses of subjects are more stable.

We used the pair comparison method because our experiment deals with very small differences in quality. The pair comparison method is more sensitive than the double stimulus continuous quality scale (DSCQS) method used in [31]. DSCQS requires subjects to mark both videos, then DMOS is calculated to do the comparison. Pair comparison, however, asks subjects to mark the difference between two videos directly. It is known to work better for very subtle differences. Since the rating includes the option of "the same," it requires fewer subjects than forced choice when the purpose is to show that two videos are subjectively the same. The rating scale does not bias subjects as does degradation category rating [11], which assumes that the test video has lower quality than the reference.

In our experiment, each video clip was 10 seconds long. Long sequences can produce a "forgiveness" effect, in which users forget and forgive quality lapses which occurred early on. One second of gray screen was shown between the videos in each paired comparison. Our videos all have spatial resolution of 1920×1080. The video clips used had a range of content: high motion and low motion, as well as content which is spatially simple and spatially complex.

3.3.3 Subjective Test

The test was held in a room with typical office lighting conditions. We included 10 test sequences. There are 3 viewing modes and 3 pairs to be compared in each mode. Therefore, we had 90 pairs to be shown in total, excluding null tests. Each pair was compared by 15 observers. Thirty subjects (20 male, 10 female, average age 25.2 years) participated in the test. Each subject compared 45 pairs of test videos and 6 null tests. After the experiment, a playback problem was found with one sequence (the playback of

the *High* version was jerky, leading it to be liked less than *Low*) so this sequence (not included in Table 3.1) was excluded from our data analysis. An experimental session was divided into six parts, where the modes were In-Hand, On-Lap, On-Stand, In-Hand, On-Lap and On-Stand. In each of the first 3 parts, subjects compared 8 pairs, and in the last 3 parts, they compared 9 pairs. There was one null test randomly placed in each part. After the 2nd and 4th parts, subjects were asked to take a break.

Written user instructions were provided at the beginning to each subject. The instructions described the three viewing modes, the experiment procedure, the grading scale and the interface. The three viewing modes were demonstrated by the experimenter. The subject then did a practice run (using unrelated sequences) to become familiar with the experiment procedure. The whole experiment took about 40 minutes.

3.4 Results and Discussion

From the scores provided by the subjects, we use a one-sided test because in each case if the difference is not zero, there is a clear direction in which we would expect the difference to lie. The null hypothesis is that the mean score μ is equal to 0, i.e., the compared pair has the same subjective quality. For different comparisons, our alternative hypotheses are selected as: (1) *UAV-High*: $\mu < 0$, (2) *UAV-Low*: $\mu > 0$, (3) *High-Low*: $\mu > 0$.

The ideal result for this experiment would be that: for *UAV-High*, we cannot reject the null hypothesis that the tested pair has the same subjective quality; and for *UAV-Low* and *High-Low*, we can reject the null hypothesis. We use a one-sided test because it would be significant for us if *UAV* has lower quality than *High*, and if *Low* has lower quality than *UAV* and *High*.

The results of t-tests for each comparison in each viewing mode are in Table 3.2.

Table 3.2: Results of t-test for data from all the subjects

Mode	*UAV-High*	*UAV-Low*	*High-Low*
Hand	fail to reject	reject	reject
Lap	$p = 0.06$	fail to reject	reject
Stand	reject	fail to reject	fail to reject

The table has "fail to reject" when $p > 0.1$ and "reject" when $p < 0.01$. We give the p-value in Table 3.2 if $0.01 < p < 0.1$. We also plot the means and 95% confidence intervals (CIs) in Fig. 3.3.

Table 3.2 shows that all comparisons of *UAV, High, Low* in In-Hand mode correspond to the ideal result. The null hypothesis of (*UAV-High*) cannot be rejected, and the null hypothesis of (*UAV-Low*) and (*High-Low*) can be rejected.

On-Lap mode: Table 3.2 shows that the null hypothesis of both (*UAV-High*) and (*UAV-Low*) cannot be rejected (though the p-value of *UAV-High* is marginal), which may indicate that no difference was observed among the three. However, when *High* was compared with *Low*, subjects seemed to notice the difference between them as the null hypothesis is rejected. So there is an inconsistency here.

On-Stand mode: the null hypothesis of (*UAV-High*) can be rejected, whereas the null hypothesis of (*UAV-Low*) and (*High-Low*) cannot. Again there is an inconsistency.

When we check the CIs of the null tests, we find that the CI of the null test in In-Hand mode unexpectedly does not include 0. There are relatively fewer of the null tests than there are of the other comparisons. Some subjects reported anecdotally after the experiment that a large number of sequences were very similar, and that it was hard to find differences. This difficulty is to be expected, since the test was designed to see whether video versions which were designed to be visually equivalent were in fact visually equivalent. It may be that the paucity of clear differences led viewers to sometimes find differences when there were none.

Given these observations, we examine subject reliability in more detail.

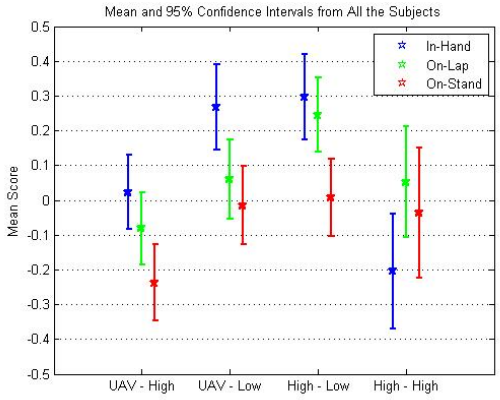


Figure 3.3: Mean scores and CIs from all the subjects

3.4.1 Analysis of Null Tests

The histogram of the number of subjects who reported a difference when none existed is shown in Fig. 3.4. It shows, for example, that only six subjects out of 30 did not report any difference on any of their null tests. Ten out of 30 subjects reported differences on two or more null tests, and six out of 30 subjects reported differences on three or more null tests. Their data may be less reliable.

To check for fatigue, we looked at whether subjects are more likely to report difference in the null tests as they watch more videos. Table 3.3 shows the fraction of subjects who reported no difference in the jth null test. As mentioned before, the first and fourth parts are In-Hand, the second and fifth parts are On-Lap, and the third and sixth are On-Stand. After the second and fourth parts, the subjects were notified to take a break. Table 3.3 shows that the subjects are slightly more likely to give accurate scores at the beginning of the experiment and after breaks. For example, 77.3% of the subjects reported no difference in the first null test, while only 51.9% reported no difference in the fourth null test (the second In-Hand part). In the On-Lap parts, more subjects reported no difference in the fifth part which followed a break, than in the second part. On-Stand is similar, with slightly higher correctness in the third part than in the sixth part.

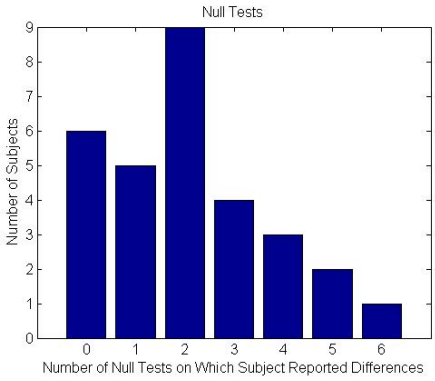


Figure 3.4: Histogram of numbers of subjects who reported difference on null tests

Table 3.3: Fraction of subjects who did not report difference in each null test.

Mode	First Null Test		Second Null Test	
	Part No.	Correct%	Part No.	Correct%
Hand	1	77.3%	4	51.9%
Lap	2	62.1%	5	65.5%
Stand	3	58.6%	6	50.0%

3.4.2 Results from Reliable Subjects and Reliable Parts

As the null tests show that some subjects are more reliable than others, and some parts may have more of a fatigue effect, we re-analyze the data from reliable subjects (reported difference in at most two null tests) and from the more reliable parts of the experiment (first part of the experiment for In-Hand mode, fifth part for On-Lap, third part for On-Stand). The fraction of subjects who reported no difference in null tests in those 3 parts is 95%, 90% and 75%.

Table 3.4 shows the results of t-test of the reliable data. We plot the means and 95% CIs in Fig. 3.5. The results change slightly from the previous results which used all data.

In-Hand mode: as before, the null hypothesis of (*UAV-High*) cannot be rejected, and the null hypothesis of (*UAV-Low*) and (*High-Low*) can be rejected with strong evidence, corresponding to the ideal result.

Table 3.4: p-values of t-test for data from reliable parts and subjects

Mode	*UAV-High*	*UAV-Low*	*High-Low*
Hand	fail to reject	reject	reject
Lap	fail to reject	fail to reject	reject
Stand	$p = 0.03$	fail to reject	fail to reject

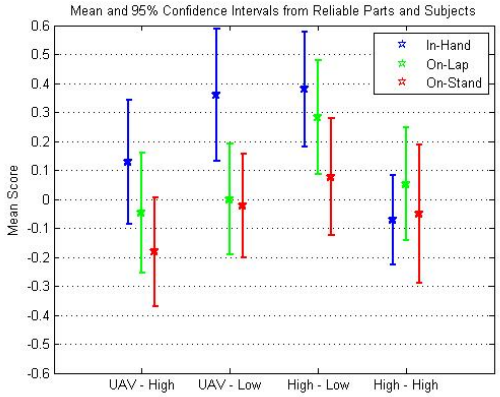


Figure 3.5: Mean scores and CIs from reliable parts and subjects

On-Lap mode: the data shows that we cannot reject the null hypothesis of both (*UAV-High*) and (*UAV-Low*), but we can reject the null hypothesis of (*High-Low*). The p-value of *UAV-High* is no longer marginal. So there is more of an inconsistency than before.

On-Stand mode: the null hypothesis of (*UAV-Low*) and (*High-Low*) cannot be rejected, while the null hypothesis of (*UAV-High*) is on the margin. If we take 0.01 as the significance level, the null hypotheses of the three comparisons cannot be rejected, which means we cannot exclude that the three versions have the same subjective quality. If we take 0.05 as the significance level, the result shows inconsistency.

3.4.3 Discussion

The subjective visual quality of a high rate encoding of original content was compared with an encoding at a lower rate and with encoding content pre-filtered for the

anticipated viewing conditions.

For In-Hand mode corresponding to the shortest viewing distance (most demanding viewing conditions), the visual quality of the *Low* version was worse than both the *High* version and the *UAV* version. For this mode, the 3% bit-rate savings of *UAV* did not degrade perceptual quality, but the attempt to realize 40% rate savings with *Low* results in visibly reduced quality.

For the intermediate case of On-Lap, the results are inconclusive but suggest that the pre-filter may be able to save on average 29% of the bit rate without degrading perceptual quality. The *Low* version is also not equivalent to the *High* version for this mode.

At the longest viewing distance (least demanding viewing conditions) of On-Stand, the results are inconsistent when using all data. When using the data from reliable subjects and parts, the data suggest that all three versions (*High*, *Low*, and *UAV*) may be perceptually equivalent. It would be important to ascertain whether the distance people use for the On-Stand mode is actually the distance for which the filtering was intended.

The videos in the experiment had subtle differences. Some subjects reported that the test was frustrating because so many videos looked equal. Many subjects could not reliably identify identical videos as being identical (nonzero scores in the null tests). We suspect that this fact and the previous one are related, in that some subjects did poorly in the null tests because the experiment overall aimed at barely visible differences, and so the subjects were scrutinizing for any possible difference.

3.5 Summary

We present a subjective test that demonstrates the bit-rate reduction by adapting to viewing distance without degrading the perceptual quality. We tested three viewing

modes which correspond to three viewing distances. Average rate savings of 3% in critical In-Hand viewing and 30% approximately in an intermediate On-Lap usage modes without impacting the subjective quality were supported. Specifically, for the In-Hand and On-Lap versions, the video with pre-filtering is statistically equivalent to the video without pre-filtering *High*, but the pre-filtered video has lower bit-rate. Since the bit-rates were selected manually, it is possible that the actual bit-rate savings could be larger than what we tested. The particular tests used H.264 as the video encoder but this method of reducing video bit-rate based on adapting to viewing conditions is independent of the codec technology. The benefits of adapting to the viewing conditions are expected to be enjoyed by a range of video encoding technologies.

Acknowledgment

Chapter 3, in part, is a reprint of material as it appears in Q. Song, P. C. Cosman, M. He, R. Vanam, L. J. Kerofsky, and Y. A. Reznik, "Subjective quality of video bit-rate reduction by distance adaptation", *International Workshop on Video Processing and Quality Metrics for Consumer Electronics*, Feb. 2015. The dissertation author was the primary author and the co-author Prof. Cosman directed and supervised the research which forms the basis for Chapter 3. The co-author M. He helped with the subjective tests. The co-authors Dr. Vanam, Dr. Kerofsky and Dr. Reznik also contributed to the ideas in this work.

Chapter 4

Efficient Perceptual Enhancement Filtering for Inverse Tone Mapped High Dynamic Range Videos

The videos discussed in Chapter 2 and 3 are traditional 8-bit low dynamic range (LDR) videos. The displays are also LDR whose screen brightness for white is only several hundred nits (cd/m^2). In this chapter, we discuss a new video format, high dynamic range (HDR) videos, which has attracted considerable attention recently. We design a perceptual enhancement filter for inverse tone mapped HDR videos. Banding artifacts and blocky artifacts in the inverse tone mapped HDR videos can be greatly reduced by this efficient filter. We ran subjective tests to demonstrate the performance of the proposed filter.

4.1 Motivation

As mentioned in Chapter 1, HDR videos are represented with 12+ bit depth. The range of brightness and the color gamut are larger than the 8-bit LDR videos. The screen brightness for white of HDR displays can be over 1,000 nits. HDR displays can show darker blacks and brighter whites, leading to more details in the displayed images.

Despite the interest in HDR videos, the current distribution of videos is mostly at 8-bit depth. Although many cameras nowadays can capture 12-bit, or even 16-bit videos, videos are quantized to 8 bits for compression and distribution. Videos are also tuned for LDR displays which, for example, use gamma encoding [21] and Rec.709 [32] color space. To watch the videos on a 1,000+ nits HDR display, one needs to apply some inverse tone mapping operator (iTMO)[33, 8] to the LDR videos. This mapping is called inverse tone mapping because usually the mapping from HDR content to LDR content is called tone mapping [8]. The iTMO may not be linear. For example, highlights and light sources in an image may be expanded more than other pixels. The iTMO can also include an electro-optical transfer function (EOTF) [7] conversion and color space conversion, if the HDR display uses different EOTF (e.g., Perceptual Quantizer [7]) and color space (e.g., DCI-P3 [34]). Some iTMOs have been investigated in [35, 36, 37, 38]. Because it can be content dependent, many distributors send iTMO as metadata to help the 1,000+ nits HDR displays convert the LDR content to HDR content.

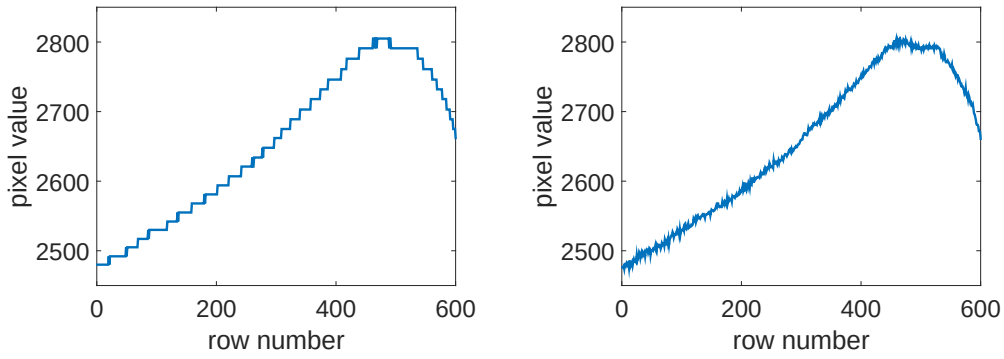
HDR video generated by iTMO sometimes suffers from false contours, also referred to as banding artifacts or ringing artifacts. The artifacts are due to the Mach band effect [39, 40], in which the human visual system enhances step boundaries by undershooting or overshooting at each step boundary. The artifacts occur especially when iTMO is a one-to-one mapping function, because 8-bit LDR video has at most 256 codewords (there are only 220 codewords in Rec. 601 [41] and Rec. 709 [32]), and after

Figure 4.1: Banding example

mapping, the HDR video also has 256 codewords. According to [7], 12-bit data, i.e., 4096 codewords, are necessary to show a banding-free image on a 1,000+ nits display. Lack of codewords in the inverse tone mapped HDR results in the visually annoying banding artifacts.

Fig. 4.1 shows an inverse tone mapped HDR image with banding artifacts around the sun in the sky. The banding is much more visible if the image is shown on a HDR display. We plot the pixel values of a column in the banding region in Fig. 4.2a. The pixel values look like staircases, and the stair height is over 10 amplitude levels (codewords) of 12-bit depth in this figure. These large steps appear as false contours.

To remove the banding artifacts, i.e., debanding or de-contouring, dithering has been used [42, 43], but the output is often not visually pleasant either (e.g, noisy in smooth regions). For stronger banding, the dithering strength has to increase, yielding noisier output. Those works aim to produce the output at the same bit depth as the input image. In our case, however, only a few codewords have been used in the HDR image



(a) inverse tone mapped HDR (b) reference 12-bit HDR

Figure 4.2: Pixel values of column 1700 of Fig.4.1.

generated by iTMO, so there is room to generate new codewords to smooth the banding artifacts. In [44], this is achieved by linear interpolation in the banding area after the banding width is identified by median filtering. The banding area can be detected by the algorithm in [45]. Some works apply lowpass filters to increase codewords. Daly and Feng [46] proposed to predict and extract false contours by lowpass filtering and quantization. The predicted false contours are subtracted from the image. This method can introduce new false ringing if the predicted false contours are inaccurate, which happens when the banding steps are non-uniform. In [47], false contours are reduced by 1D directional smoothing filters whose directions are orthogonal to the false contours. This method requires high computational complexity to detect false contours. To avoid blurring true edges, the filter size should be variable, which is hardware unfriendly. In [48], the banding region is first detected by analyzing the neighborhood at multiple scales. New codewords are generated by the expected mean value of the local neighborhood. The performance is good, but the computation is quite intense. Huang et al. [49] detect false contours by checking the eight neighbors. Several conditions are applied to exclude very smooth regions, texture and sharp edges. The contours are then removed by probabilistic dithering followed by lowpass filtering. The method was designed for removing banding

in LDR images. The conditions used in the banding detection will have to be modified for detecting banding in HDR images, and the conditions may depend on the iTMO. In [50], the bit depth is increased by taking a weighted sum of the pixels in a window, where the weights are adaptive to the content. The window size has to be large enough if this approach is applied for banding removal.

In this chapter, we propose a selective sparse filter which combines smooth region detection and banding reduction. It removes the banding artifacts, and reduces some coding artifacts, such as blocky artifacts. The properties of the inverse tone mapped HDR are exploited in the filter design. We aim at implementing the filter in hardware, so computational complexity and memory cost are carefully considered.

The overview of the system is shown in Fig. 4.3. Content tuned for LDR displays is called LDR video (i.e., the EOTF and color space of LDR displays are used to grade the content), and content tuned for HDR displays is called HDR video. The camera outputs 12-bit LDR video, which is quantized to 8 bits for compression and distribution. A legacy encoder, e.g., AVC [2] or HEVC [1], is used to encode the 8-bit LDR video. We generate the iTMO, which can be content dependent, at the encoder, and send the corresponding iTMO parameters as metadata. How to generate the iTMO is not in the scope of this work. At the decoder, the LDR bitstream is decoded and can be displayed directly on a LDR display. For HDR displays, the iTMO from metadata is applied to the reconstructed 8-bit LDR video to generate the HDR video. Although this inverse tone mapped HDR video is in 12-bit representation, it uses only 256 codewords, because the iTMO is a one-to-one mapping.

To remove banding artifacts, our proposed debanding filter is applied to the inverse tone mapped HDR video at the decoder. There are some parameters in the filter which are content dependent. The parameters are solved at the encoder side, where we have access to the 12-bit LDR video and have more computing resources. At the encoder,

a reference 12-bit HDR video is generated by applying the iTMO$^{(12)}$ to the 12-bit LDR video. The iTMO$^{(12)}$ is the higher precision version of the iTMO applied to the 8-bit LDR. The superscript (12) indicates the input bit depth is 12. This reference 12-bit HDR video is free of banding because it is represented using the full 4096 codewords. Fig. 4.2b shows the pixel values of the corresponding reference 12-bit HDR of Fig. 4.2a. The reconstructed 8-bit LDR video is also available at the encoder. To obtain the same output as the decoder side, we apply iTMO to the reconstructed 8-bit LDR video. Both the reference HDR video and the inverse tone mapped HDR video are used to select the parameters of our proposed debanding filter. The parameters are sent as metadata along with the iTMO parameters.

The rest of the chapter is organized as follows: In Sec. 4.2 we explain our proposed debanding filter, and discuss the parameter selection in Sec. 4.3. Performance evaluation and comparisons of our filter with other debanding algorithms are in Sec. 4.4, and Sec. 4.5 summarizes the chapter.

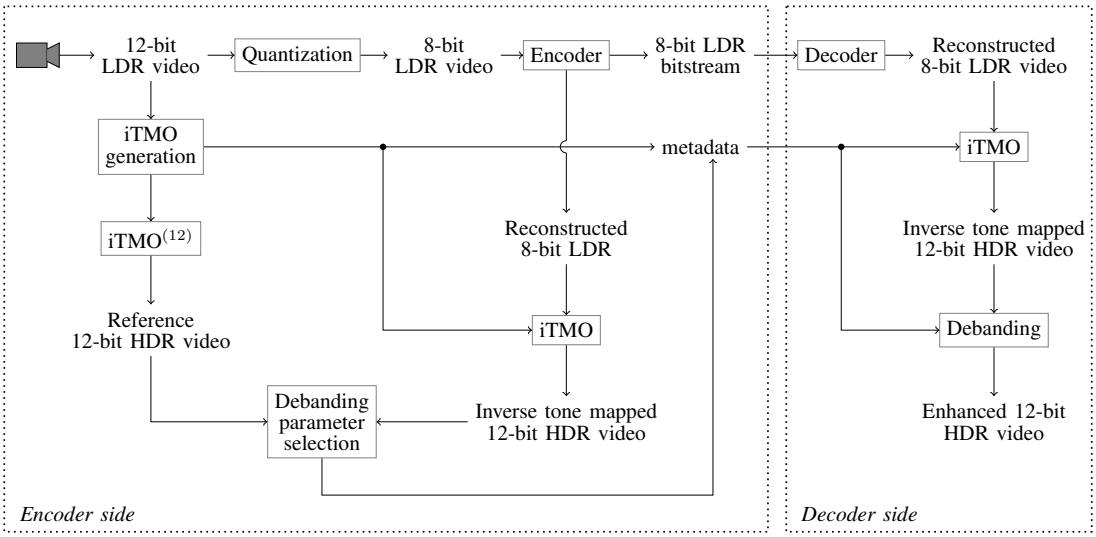


Figure 4.3: Overview of system

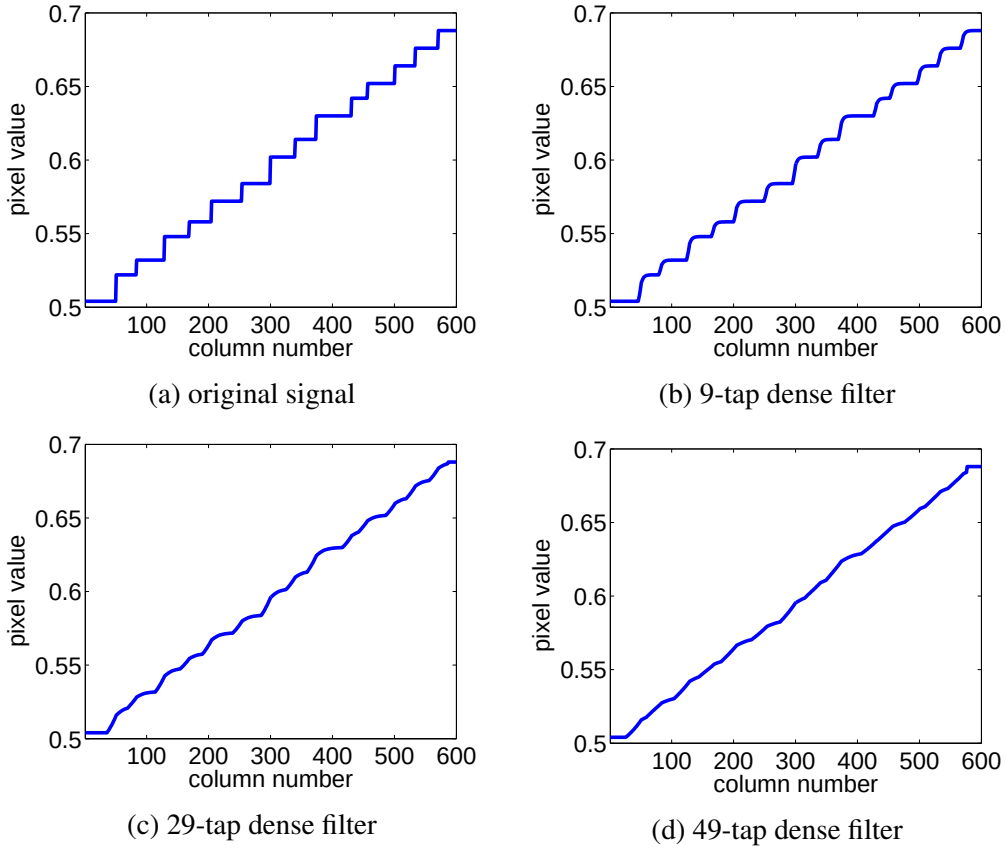
4.2 Proposed Edge-Aware Sparse Filter

Banding artifacts usually occur in regions of small gradient, and the artifacts appear as steps. We define a banding step as a group of consecutive pixels which have the same codeword, and the pixel on the left (top) and the pixel on the right (bottom) of this group have different codewords from the group. To remove the artifacts, one can smooth the area by adding more codewords between each banding step. One simple method is to apply a lowpass filter. A traditional dense 2D FIR filter can be represented as: $y[m,n] = \sum_{i=-u}^{u} \sum_{j=-v}^{v} w_{i,j} \cdot x[m+i, n+j]$, where $x[m,n]$ is the input signal at row m and column n, $y[m,n]$ is the corresponding output signal, and $w_{i,j}$ is the filter coefficient. An unweighted lowpass filter has equal coefficients: $w_{i,j} = \frac{1}{(2u+1)(2v+1)}$. This filter averages a total of $(2u+1)(2v+1)$ input pixels centered at $x[m,n]$. The 2D filtering is separable: it is equivalent to sequentially applying a 1D $(2v+1)$-tap horizontal averaging filter and a 1D $(2u+1)$-tap vertical averaging filter, which is much more efficient.

To remove banding, we have to apply a filter whose span is wide enough. Fig. 4.4a shows a 1D signal with non-uniform steps. The pixel values are normalized to $[0,1]$. Fig. 4.4b-4.4d show the outputs of the dense filter with different numbers of taps. The 9-tap filter is not able to remove the false contours; there are many wide steps left. The 29-tap filter works better. The banding is almost gone when the number of taps increases to 49.

The dense filter can smooth banding only when pixels on more than one step are involved in the averaging, even if we allow dynamic filter coefficients. When the span of the filter is not wide enough, many consecutive pixels of the output will have the same codeword, because the pixels taken for averaging are from the same banding step. If the banding step size is uniform, the false contours can be completely smoothed out when the span of the filter is $2W-1$, where W is the width of each banding step. That means,

when the step is wide, we would have to increase the span of the filter, i.e., increase the number of taps of the filter. To implement the filtering in hardware, we need to put each row of pixels into one line buffer for vertical filtering. The cost of the dense filter would be high because the filtering module would need one line buffer for each tap of the filter.



(a) original signal

(b) 9-tap dense filter

(c) 29-tap dense filter

(d) 49-tap dense filter

Figure 4.4: Performance of dense filter

4.2.1 Sparse Filter

From the observations above, we learn that to remove banding, the key is to get samples from different steps. A sparse filter does that efficiently. A 1D horizontal sparse

FIR filter [51, 52] is defined as:

$$z[m,n] = \sum_{j=-v}^{v} \hat{w}_j x[m, n+s_j], \tag{4.1}$$

where s_j is the distance from the original pixel to the sampled input signal, and $\hat{w}_j$ is the coefficient of the j-th tap. The number of taps is $2v+1$. Fig. 4.5b shows a 5-tap horizontal sparse filter with the same span as the 13-tap dense filter in Fig. 4.5a. The origin is marked by dark gray. The distances between each two neighboring samples are the same. Figs. 4.5c - 4.5f show the 5-tap horizontal sparse filtering outputs of Fig. 4.4a with different spans. The filter coefficients are fixed and equal: $\hat{w}_j = \frac{1}{2v+1}$ for all j. The outputs in Figs. 4.5d and 4.5e look relatively smooth. Figs. 4.5g and 4.5h show the outputs of 3-tap and 9-tap sparse filters with span 41, respectively. The 3-tap sparse filter creates fewer codewords than the 5-tap filter, so the output looks more jagged than Fig. 4.5e. The output of the 9-tap sparse filter is smoother than the output of the 5-tap filter, and looks similar to Fig. 4.4d. The computation is much lighter than the dense filter.

Note that we use a 1D signal as an example here. The 2D sparse filtering can be obtained by applying this filter horizontally and then vertically. The final output is:

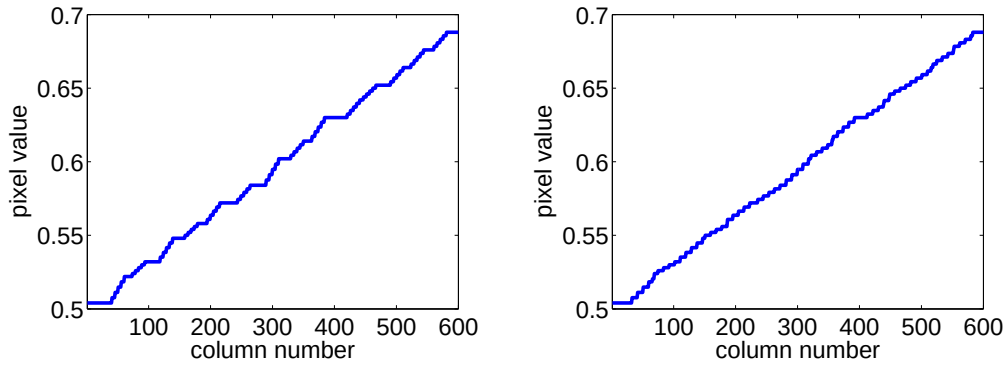
$$y[m,n] = \sum_{i=-u}^{u} \tilde{w}_i z[m+t_i, n] = \sum_{i=-u}^{u} \sum_{j=-v}^{v} \bar{w}_{i,j} x[m+t_i, n+s_j], \tag{4.2}$$

where $\bar{w}_{i,j} = \tilde{w}_i \cdot \hat{w}_j$.

The sparse filter needs fewer taps than the dense filter. A $(2u+1)$-tap vertical sparse filter with any span requires only $2u+1$ line buffers. The required memory size of the sparse filter is much smaller than the dense filter, though the sparse filter increases memory traffic. We need fewer adders and multipliers for the sparse filtering.

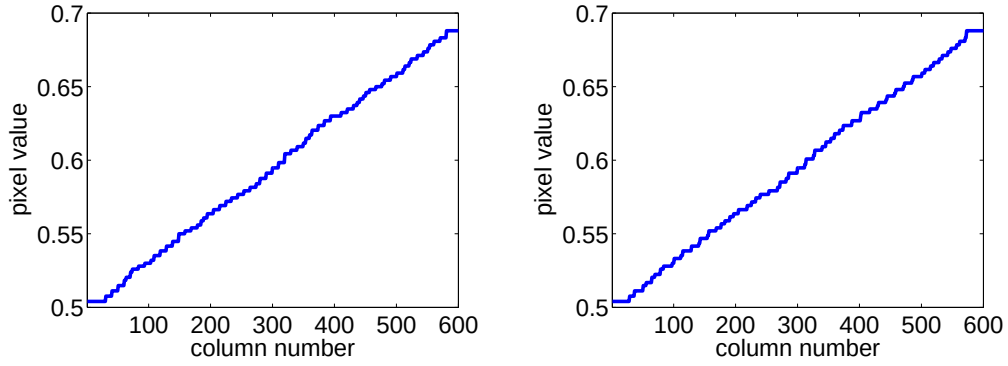
(a) 13-tap dense filter

(b) 5-tap sparse filter



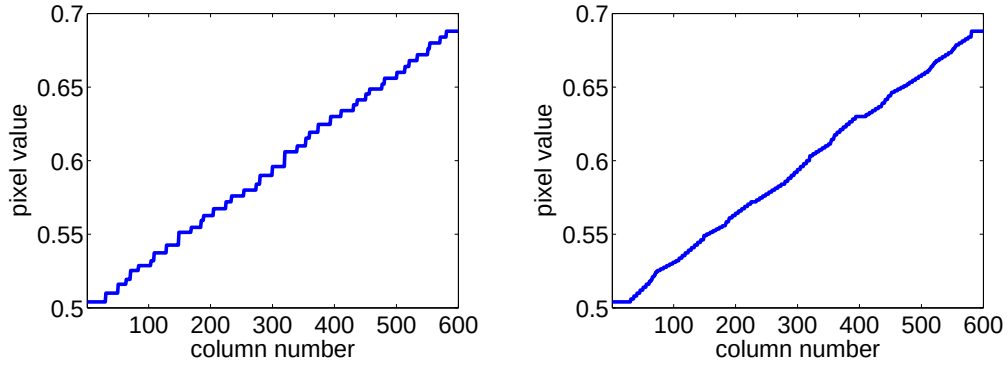
(c) 5-tap, span = 21

(d) 5-tap, span = 37



(e) 5-tap, span = 41

(f) 5-tap, span = 57



(g) 3-tap, span = 41

(h) 9-tap, span = 41

Figure 4.5: Performance of 5-tap sparse filter.

4.2.2 Edge-Aware Selective Filter

It is clear that sparse FIR filters can help remove false contours, but they can also blur true edges and remove details. To address this issue, we propose to apply the

sparse filter selectively, to smooth areas only. Banding is only observed in smooth areas. Also, smoothing a smooth area would not cause much loss of detail even if there were no banding in the area.

The proposed filter includes a horizontal filter and a vertical filter. The two filters will be applied sequentially. The horizontal filter has $2v+1$ taps. For simplicity, the corresponding vertical filter has the same structure as the horizontal filter, with the same number of taps and the same sample locations.

Fig. 4.6 shows the flowchart of our proposed horizontal filter with 7 taps as an example ($v = 3$). The input image is the inverse tone mapped HDR image. For each pixel $x[m,n]$ in the input image, we sample itself and v pixels on the left and right sides of it. The positions of the sampled pixels are denoted $n+s_j$ where $j \in \{-v,\cdots,-1,0,1,\cdots,v\}$ and $s_0 = 0$. We compute the difference between the central pixel $x[m,n]$ and each of the sampled pixels $x[m,n+s_j]$ where $j \neq 0$. If the absolute value of the difference is below a threshold Δ, we determine that the sampled pixel has a similar value to the central pixel. If the central pixel and all the sampled pixels have similar values, we consider the area to be smooth, and replace the central pixel value with the average of the inputs. The averaging takes only $2v-1$ inputs: $x[m,n+s_{-v+1}],\cdots,x[m,n+s_{-1}],x[m,n],x[m,n+s_1],\cdots,x[m,n+s_{v-1}]$. If the difference between the central pixel and any of the sampled pixels is greater than the threshold, there may be edges or texture in the area. Then the averaging is not applied, and the input pixel value remains unchanged: $z[m,n] = x[m,n]$.

For vertical filtering, v pixels on the top and bottom of the central pixel $z[m,n]$ are sampled from the horizontal filtering output. As before, the averaging is applied only when the differences between the central pixel and all the sampled pixels are within the threshold. Only $2v-1$ inputs are used for averaging. The output of the vertical filtering is denoted $y[m,n]$, which is the enhanced HDR in Fig. 4.3.

This selective filter combines non-smooth area detection and sparse filtering. The

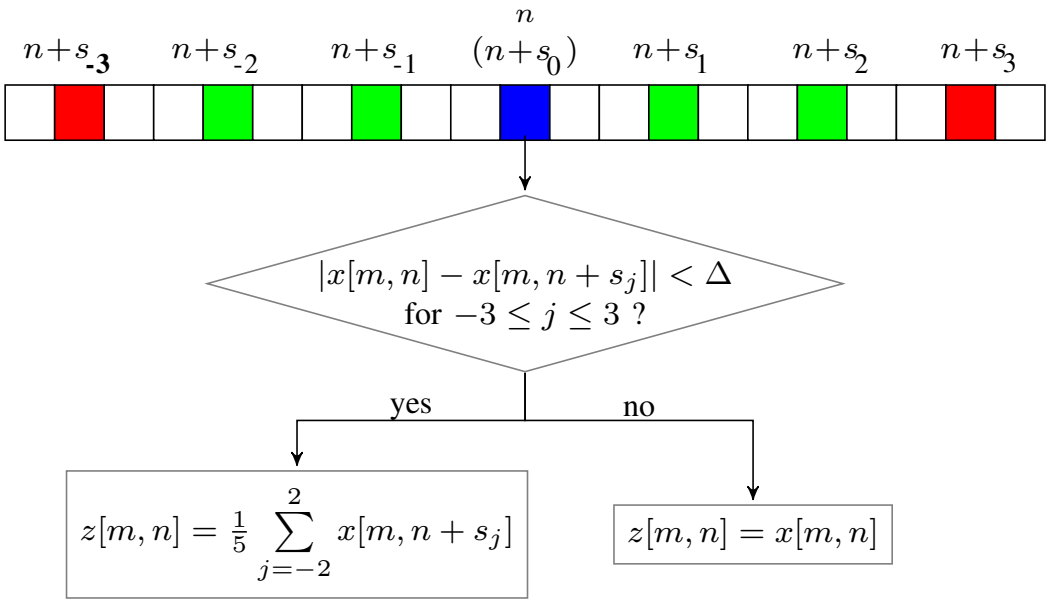


Figure 4.6: Flowchart of proposed edge-aware sparse filter

decoder will be able to make the decision whether to apply averaging to each pixel, thus no filtering map needs to be sent from the encoder. The filter only requires one line buffer for the horizontal filtering and $2v+1$ line buffers for the vertical filtering. The selection of the threshold Δ for the selective sparse filter is critical to the debanding performance. Note that the filtering process only takes $2v-1$ pixels, and the extra two pixels are for non-smooth area detection, as will be discussed below. Because banding artifacts appear in smooth regions, and would not occur near object boundaries, we do not want to involve pixels near boundaries. In the following, we first describe how to select the threshold, then explain why the extra two pixels are necessary in the decision process.

Adaptive threshold

The threshold Δ indicates how much difference we will tolerate in the decision process. If Δ is too small, we will only average pixels with small differences, corresponding typically to small areas, so banding artifacts might not be removed. If Δ is too large,

the filter could be applied to areas with sharp edges and details, leading to blurred edges and loss of detail.

One important observation of the banding areas is that the codewords of the corresponding pixels in the 8-bit LDR image are very similar to each other. The difference of the 8-bit LDR codewords between neighboring pixels is 1 or 2 most of the time. After inverse tone mapping, the difference between these neighboring pixels shown on a HDR display becomes larger, and that results in the banding artifacts. Since the input of our proposed filter is the inverse tone mapped HDR, Δ can be related to the mapping function.

Assume that the codeword of a pixel in the 8-bit LDR is b where $0 \leq b \leq 255$. The corresponding inverse tone mapped HDR codeword is $T(b)$. The difference between two neighboring HDR codewords is denoted $dT(b) = |T(b+1) - T(b)|$. In the inverse tone mapped HDR image, if an area is relatively smooth, we expect the values of nearby pixels to be around $T(b)$. If a pixel is in a textured area, the differences among nearby pixels could be much larger than that. Therefore, we set the threshold Δ to a small number times $dT(b)$.

We discuss two commonly used iTMO functions as examples. First, the iTMO is a simple linear mapping: $T(b) = \rho \cdot b + c$, where ρ is positive and constant for the entire image, and c is a constant offset. For example, ρ can be $\frac{2^{12}}{2^8} = 16$ for simple bit depth up-conversion. We set the threshold for the entire image to

$$\Delta = \alpha \cdot dT(b) = \alpha \cdot (\rho \cdot (b+1) + c - \rho \cdot b - c) = \alpha \cdot \rho, \tag{4.3}$$

where α is positive.

Another popular iTMO is the piecewise polynomial [38]. Sometimes the iTMO is non-linear, and is represented by a piecewise polynomial. With $\hat{K}$ segments in total in the iTMO curve, the differential function $dT(b)$ is partitioned into $\hat{K}$ segments. The

segment boundary points are denoted p_k, where $1 \leq k \leq \hat{K}+1$. The segment slopes can be very different, so different thresholds are needed. When the codeword of the central pixel of the filter inputs ($x[m,n]$) is $T(b)$, the threshold is set to

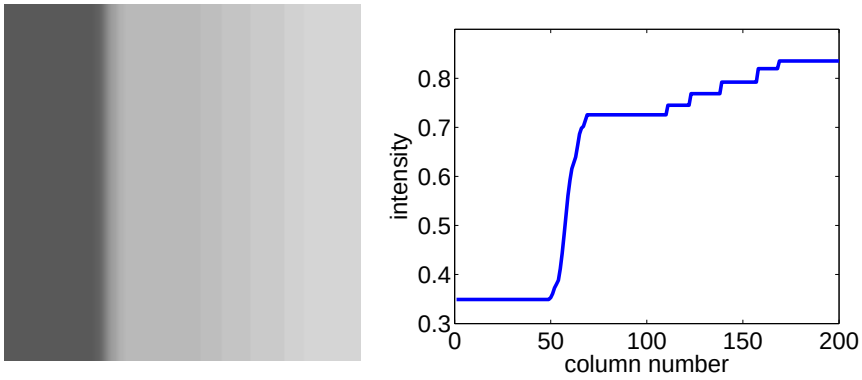
$$\Delta = f(b) = \alpha \cdot \max_{p_k \leq b < p_{k+1}} \{dT(b)\}, \tag{4.4}$$

where α is positive. The threshold is set to the maximum differential of the segment multiplied by a factor α. In our tests, for both the linear mapping and the piecewise polynomial, $\alpha = 2$ or 3 usually works well.

This method can be extended to other iTMO algorithms. The differential function $dT(\cdot)$ of any one-to-one mapping can be built. The point is that we only allow averaging a few codewords in the filtering process. So the threshold can be set to $\alpha \cdot dT(b)$ when $x[m,n] = T(b)$. Another possible setting for the threshold is α times the maximum codeword differential of the entire image: $\alpha \cdot \max_{0 \leq b < 255}\{dT(b)\}$.

Extra samples for non-smooth area detection

We include $2v+1$ pixels for decision but only take $2v-1$ pixels for filtering. This prevents introducing new false ringing to the output image. We use the patch in Fig. 4.7a to illustrate; it has an edge between the dark and bright regions, and banding artifacts in the bright region. We plot the intensities of a row of pixels in Fig.4.7b. Our goal is to preserve the dark region and the edge, and smooth the banding in the bright part. Using a 7-tap ($v = 3$) filter as an example, we set the filter parameters $s_1 = -s_{-1} = 7, s_2 = -s_{-2} = 14, s_3 = -s_{-3} = 17$, and set $\Delta = 2H$ where H is the maximum difference between adjacent banding steps. Figs. 4.8a and 4.8b show the mid region of Fig. 4.7b. We want to determine whether to apply the sparse filter to the pixels marked by blue circles. The range of $[x[m,n]-\Delta, x[m,n]+\Delta]$ is marked by dashed lines.



(a) (b)

Figure 4.7: Example of banding artifacts

We apply the filter only when all the six samples have values similar to the central pixel. For the pixel marked by the blue circle in Fig. 4.8a, the difference between it and the leftmost sample exceeds the threshold, so the filtering is not applied. For the pixel marked by a blue circle in Fig. 4.8b, all the samples marked by green crosses and red triangles are within the threshold. We apply the sparse filtering by averaging only the five central pixels, not all seven pixels. So we exclude the leftmost sample which is an outlier from the averaging. The filtering result is shown in Fig. 4.8c. The banding artifacts are smoothed, and the edge is well preserved.

If we do not get the two samples marked by red triangles, and determine to apply the filtering as long as the four green cross samples have similar values to the central pixel, a false ringing can be introduced. For the pixel marked by the blue circle in Fig. 4.8a, the filtering condition would be satisfied. However, the leftmost green cross sample is actually at a transition area. That sample is an outlier, whose value is slightly different from the others though the difference is still within the threshold. The average of the five pixels would be slightly lower than the original value which brings an undershoot (marked by the solid red circle in Fig. 4.8d). Similarly, an overshoot is introduced on the other side of the true edge (marked by the dashed green circle in Fig. 4.8d). In the image, the overshoot and undershoot appear as faint false ringing.

Therefore, with the extra two samples to probe if there is an edge nearby, we prevent introducing new artifacts into the output image. Also, more details can be preserved with the extra two samples strengthening the condition to apply filtering.

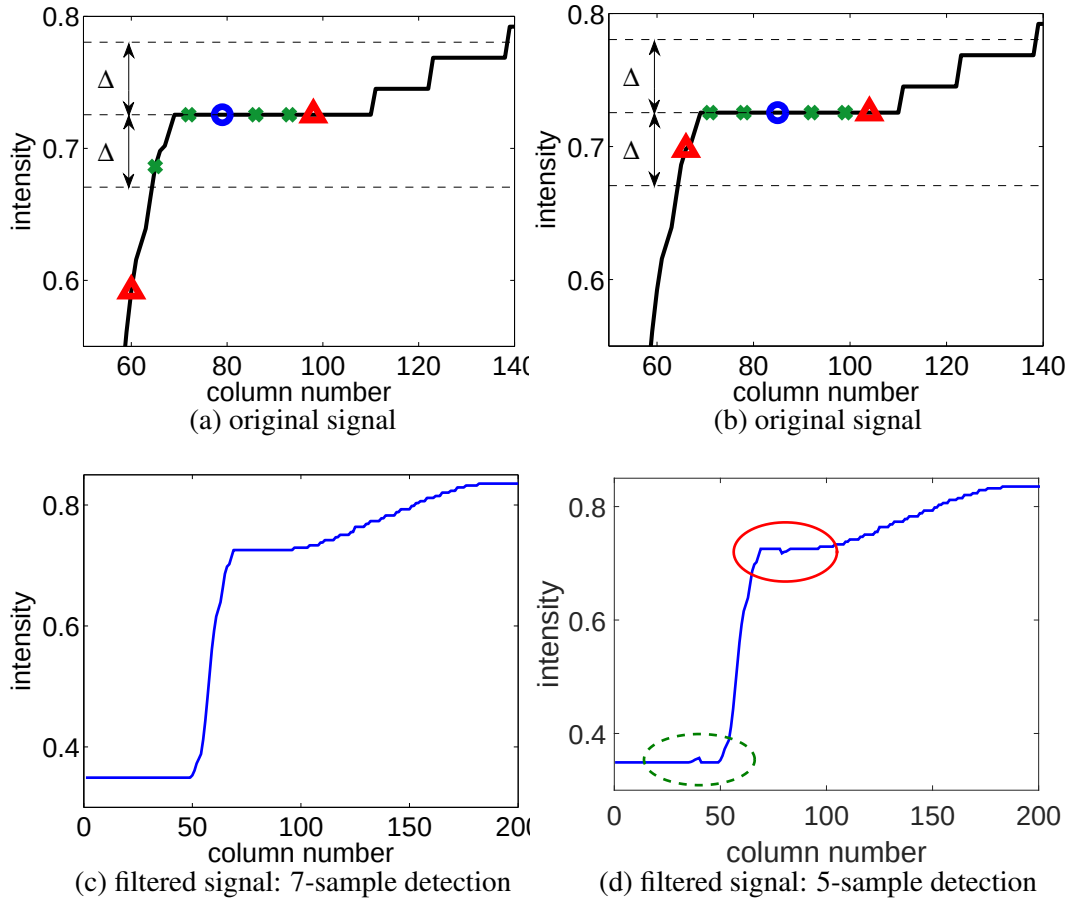


Figure 4.8: Comparison between 5-sample and 7-sample non-smooth area detection

Metadata

This selective sparse filter is to be applied at the decoder side and implemented in hardware. The number of line buffers used for filtering needs to be fixed, so the number of taps of the filter has to be fixed. We found that 7 taps in total (i.e., 5 taps for averaging) are usually enough for the sparse filter to remove the banding artifacts. If the banding steps are uniform, a 5-tap 1D sparse filter (no decision process) can create at most 4 new

codewords at each banding step (see Appendix). If the banding steps are in 2D and are uniform, we can create at most 16 codewords after applying the sparse filter in both the horizontal and vertical directions. That means the bit depth is increased by 4 (from 8 to 12). If the banding steps are non-uniform, it is possible to create more codewords.

There are several parameters to be determined according to frame content: 1) α in the threshold, and 2) the positions of samples in the sparse filter. In our tests, we found that equidistant samples for averaging works well, i.e., we set $s_j = -s_{-j} = jD$ for $1 \leq j \leq v-1$. For the extra two samples for non-smooth area detection (s_v and s_{-v}), empirically we set $s_v = s_{-v} = \lfloor \frac{2v-1}{2} s_1 \rfloor = \lfloor \frac{2v-1}{2} D \rfloor$, i.e., the distance between $x[m, n+s_v]$ and $x[m, n+s_{v-1}]$ is half of the distance between $x[m,n]$ and $x[m, n+s_1]$. The span of the filter in the averaging process is $2(v-1)D+1$, and the entire span of the filter in the decision process is $2(v-1)D+1+2\lfloor \frac{D}{2} \rfloor$.

In summary, the filter parameters to be determined are

- the threshold factor: α,
- the distance between each two neighboring samples for averaging: D.

The metadata is simple. There is no need to store or transmit a filtering map. A set of parameters is to be determined to smooth all the banding in the image. As can be seen from Fig. 4.5, the span of the sparse filter is critical. Increasing the span may not always make the signal smoother. In the next section, we will describe how to select the parameters using the reference 12-bit HDR video and the inverse tone mapped HDR video.

4.3 Parameter Selection

At the encoder, we have more computational resources, and access to the reference 12-bit HDR video. So we can filter with different spans and thresholds, compare with

the reference, and select the parameters that yield the best output. One may think that the output which has the minimum distortion from the reference 12-bit HDR video is the best. However, simple metrics, such as mean squared error (MSE) and SSIM [9], are usually not consistent with the visual quality. Therefore, we propose a new metric to measure the perceptual quality of the filtered image. We then formulate the problem to optimize the parameters.

4.3.1 Perceptual Distortion

Generally, we consider two aspects when measuring the quality: (a) how well the banding is smoothed, and (b) how well the true edges and details are preserved.

Smoothness after Filtering

The pixel value at row m and column n in the inverse tone mapped HDR is denoted $x[m,n]$. We can find banding steps in the inverse tone mapped HDR image in the horizontal and vertical directions individually by two raster scans. We denote a horizontal banding step in row m_0 from column n_1 to n_2 as $\Omega_i = \{m_0, n_1, n_2\}$, where $x[m_0,n]$ is the same for $n_1 \leq n \leq n_2$, and $x[m_0,n_1-1] \neq x[m_0,n_1]$, $x[m_0,n_2+1] \neq x[m_0,n_2]$. The width of Ω_i is denoted L_i^H, where $L_i^H = n_2 - n_1 + 1$. Similarly, we denote a vertical banding step in column n_0 from row m_1 to m_2 as $\Phi_j = \{n_0, m_1, m_2\}$, where $x[m,n_0]$ is the same for $m_1 \leq m \leq m_2$, and $x[m_1-1,n_0] \neq x[m_1,n_0]$, $x[m_2+1,n_0] \neq x[m_2,n_0]$. The width of Φ_j is denoted L_j^V, where $L_j^V = m_2 - m_1 + 1$.

In the averaging process, our proposed debanding filter is a $(2v-1)$-tap sparse filter with fixed equal filter coefficients. The banding steps will be broken into many mini-steps after sparse filtering. We observe that the widest mini-step width after filtering shows how much banding remains. The filtering output is denoted as $y_{D,\alpha}[m,n]$ when D and α are used. The widest width of output mini-steps of Ω_i is denoted $l_i^H(D,\alpha)$, and

that of Φ_j is denoted $l_j^V(D,\alpha)$.

We define a *residual banding ratio* of horizontal banding as $r_i^H(D,\alpha) = \frac{l_i^H(D,\alpha)}{L_i^H}$. Similarly, a residual banding ratio of vertical banding is defined as $r_j^V(D,\alpha) = \frac{l_j^V(D,\alpha)}{L_j^V}$. We pool the residual banding ratio of all the banding steps in the image, and compute the *residual banding level* of the whole image as

$$ResB(D,\alpha) = \frac{\sum_i r_i^H(D,\alpha)\cdot L_i^H + \sum_j r_j^V(D,\alpha)\cdot L_j^V}{\sum_i L_i^H + \sum_j L_j^V}. \tag{4.5}$$

We weight the residual banding ratio of each banding step with the width of the banding because wide banding is usually more visible than narrow banding. The pooling is normalized by the sum of weights: $\sum_i L_i^H + \sum_j L_j^V$. After simplification, we obtain:

$$ResB(D,\alpha) = \frac{\sum_i l_i^H(D,\alpha) + \sum_j l_j^V(D,\alpha)}{\sum_i L_i^H + \sum_j L_j^V}. \tag{4.6}$$

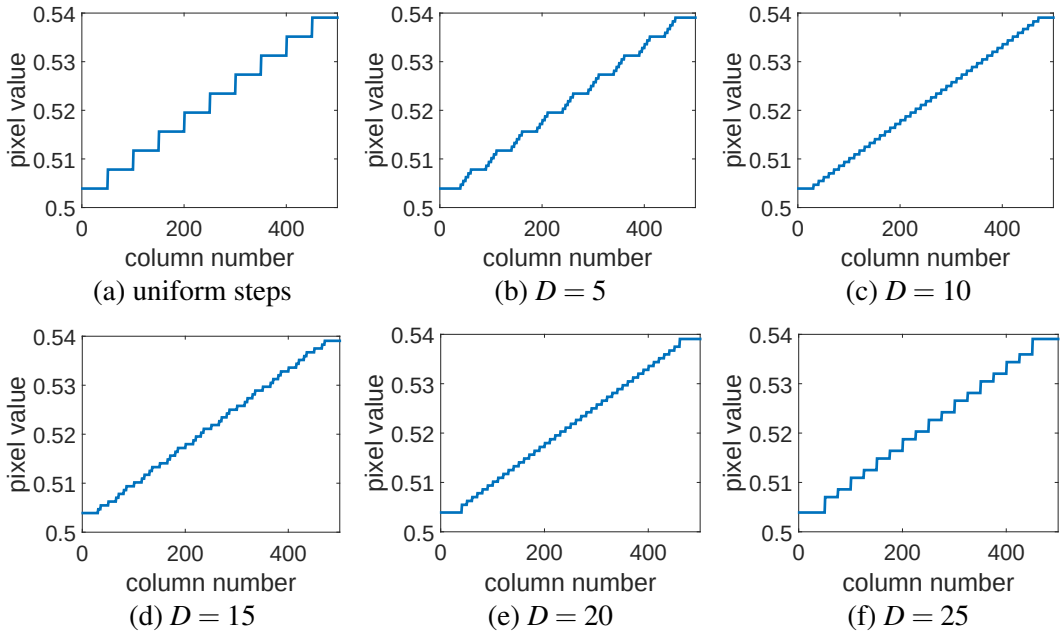
Note that $0 < ResB(D,\alpha) \leq 1$. Smaller $ResB(D,\alpha)$ means the output is more smoothed.

In the implementation, we exclude banding steps:

- where the 12-bit reference HDR is also flat (pixels in the region have the same codeword);
- the first and the last step of consecutive banding steps. If there are only two consecutive steps in the group, then remove the longer step;
- which are shorter than B pixels. For image resolution 1920×1080, we set B to 7, and for resolution 3840×2160, we set B to 14.

The remaining steps are called *major* banding steps. If there are no major banding steps, i.e., $\sum_i L_i^H + \sum_j L_j^V = 0$, we set $ResB(D,\alpha)$ to 0.

To show the relationship between *ResB* and the smoothness of the filtered output, we first consider 1D synthesized data with steps of uniform width and uniform codeword



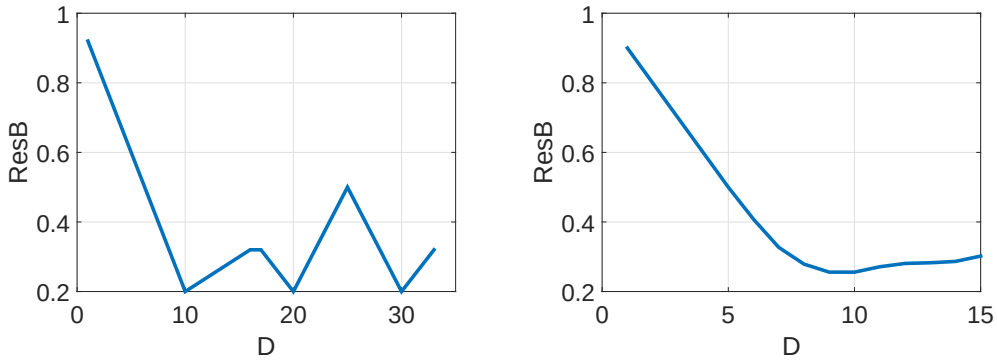
(a) uniform steps (b) $D = 5$ (c) $D = 10$

(d) $D = 15$ (e) $D = 20$ (f) $D = 25$

Figure 4.9: Uniform steps and filtering outputs with different D

difference between adjacent steps. Fig. 4.9a shows uniform steps with width $W = 50$. We plot the sparse filtering outputs using different filter spans in Fig. 4.9b-4.9f. The number of taps for averaging is 5 (i.e., the total number of taps is 7 with the two extra samples for decision). Each banding step in the input data is divided into at most 5 mini-steps. The relationship between D and the widths of mini-steps is in Appendix A.

In Fig. 4.9, the smoothest output is from $D = 10$ and $D = 20$, where the output mini-step width is uniform. When $D = 5$ (Fig. 4.9b), the widest mini-step after filtering is 30, and the banding is not smoothed well. When $D = 15$, the widest mini-step width after filtering is 15, which is larger than the widest mini-step width when $D = 10$ (Fig. 4.9c), and we can observe from Fig. 4.9d that $D = 15$ yields more jagged output. When D increases to 25, the widest mini-step width after filtering is 25, and the output (Fig. 4.9f) is coarser than the other outputs. We plot the residual banding level created by different D in Fig. 4.10a. The residual banding level is consistent with the jaggedness of the filtering output.



(a) *ResB* of Fig. 4.9a (b) *ResB* of Fig. 4.4a.

Figure 4.10: Residual banding level of filtering outputs of uniform banding steps in Fig. 4.9a and non-uniform banding steps in Fig. 4.4a

If the boundaries are ignored, the minima of the residual banding level are obtained when $D = \frac{K'W}{5} + KW$, where K' is a positive integer and is not a multiple of 5, and K is a positive integer (see Appendix A). Averaging different combinations of input codewords may result in the same output, so there are more than one minima of *ResB*. This provides the possibility to achieve the minimum residual banding level for multiple groups of banding steps in an image where the widths of each group are different.

For the example of non-uniform banding steps in Fig. 4.4a, we plot the residual banding level of the filtering output in Fig. 4.10b. The minima are at $D = 9$ and $D = 10$, which correspond to span 37 and 41 in Fig. 4.5. We can see that Figs. 4.5d and 4.5e indeed look much better than Fig. 4.5c, and slightly better than Fig. 4.5f. We assume Δ is big enough so that all the pixels are filtered.

For real data, the banding steps are usually non-uniform. To preserve the decoder hardware efficiency and limit the metadata bit overhead, we do not allow changing the sparse filter span within one image.

Fidelity to the Reference HDR

Measuring the smoothness after filtering may not be sufficient to represent the overall quality. Detail preservation should be considered. We measure the distortion between the filtering output and the reference 12-bit HDR over the whole image, including both banding and non-banding regions. MSE is used to measure the distortion for simplicity. We denote the reference 12-bit HDR image as $\hat{x}[m,n]$. The distortion is computed as

$$MSE(D,\alpha) = \frac{1}{MN}\sum_{m=1}^{M}\sum_{n=1}^{N}(y_{D,\alpha}[m,n]-\hat{x}[m,n])^2, \tag{4.7}$$

where M and N are the image height and width, respectively. Smaller MSE means higher fidelity to the reference HDR.

4.3.2 Problem Formulation

We want to reduce both the residual banding level and the MSE. We define the perceptual distortion $J(D,\alpha) = MSE(D,\alpha) + \lambda \cdot ResB(D,\alpha)$, where λ is a weighting factor that controls the trade-off between smoothness and fidelity. We select the filter parameters by

$$\{D^*,\alpha^*\} = \underset{\{D,\alpha\}\in\mathcal{D}\times\mathcal{A}\cup\{0,0\}}{\arg\min} MSE(D,\alpha) + \lambda \cdot ResB(D,\alpha), \tag{4.8}$$

where $\mathcal{D}$ and $\mathcal{A}$ are pre-defined sets of available candidates of D and α. $\{D,\alpha\} = \{0,0\}$ means the filter is not applied. MSE is computed between the reference 12-bit HDR ($\hat{x}[m,n]$) and the inverse tone mapped HDR ($x[m,n]$). The residual banding level, $ResB(0,0)$, is 1 if $\sum_i L_i^H + \sum_j L_j^V > 0$; otherwise, $ResB(0,0) = 0$.

We find the filter span and the threshold factor by minimizing the weighted sum of the two terms. This formulation is commonly used in denoising and image enhancement

algorithms [53, 54, 55, 56, 57]. This optimization is applied to each frame individually. We found that determining an optimal set of parameters for each scene does not smooth out all the banding, because the banding width can be changing over a scene, especially in fade-in / fade-out scenes.

4.3.3 Computational Complexity

At the decoder, our proposed filter needs 6 comparisons in the decision process, and one multiplication and 4 additions in the averaging process, for each pixel. A comparison needs two additions. In total, our proposed filter demands one multiplication and 16 additions for each pixel.

At the encoder, each pixel has to be compared with its top and left neighbors to determine the vertical and horizontal banding steps, respectively. For each $D \in \mathcal{D}$ and each $\alpha \in \mathcal{A}$, the proposed filter is applied once. To compute the residual banding level, each pixel is compared with its top neighbor at most once and with its left neighbor at most once. Computing MSE costs one multiplication and two additions for each pixel. For $\{D, \alpha\} = \{0, 0\}$, only MSE is computed. In total, the parameter selection requires $(2|\mathcal{D}||\mathcal{A}| + 1)$ multiplications and $(22|\mathcal{D}||\mathcal{A}| + 6)$ additions for each pixel.

4.4 Performance Evaluation

We verified our proposed filter using 4 video sequences and 8 images extracted from 5 video clips. All the 8-bit LDR videos are compressed using HEVC at 5.2 Mb/s. The resolution is 1920×1080. The number of filter taps is set to 7, and $\lambda = 10^{-5}$ in (4.8) for all the sequences and images. We provide the encoder with 2 options of α and 8 options of D. We apply the filter in the YCbCr color space. The EOTF of the inverse tone mapped HDR is Perceptual Quantization (PQ) [7]. Only the luma channel is filtered.

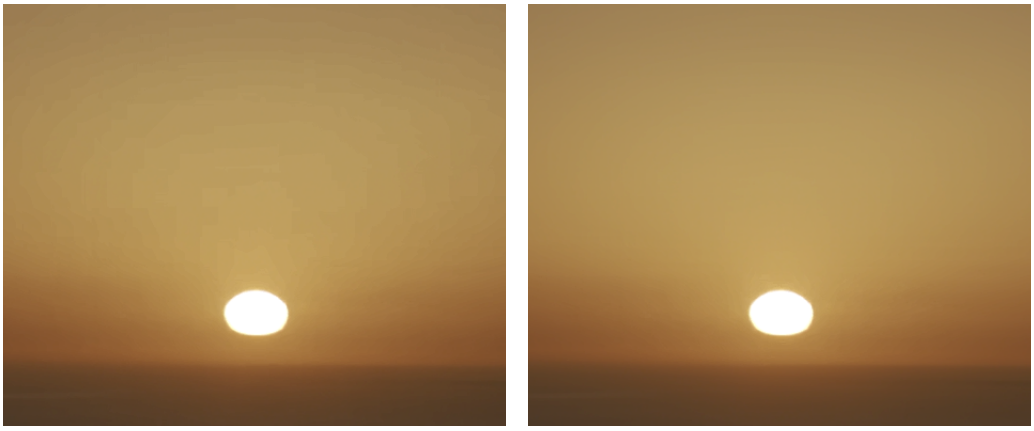
Figure 4.11: Filtered (enhanced) 12-bit HDR.

Note that this filter can be applied to a single color component (e.g., luma), or more components (e.g., chroma). It can be also applied to any inverse tone mapping with any EOTF.

The filtering output of Fig. 4.1 is shown in Fig. 4.11. We crop a patch around the sun where the banding is severe, and show it in Fig. 4.12. The banding is smoothed out, and the edges and details are well preserved. We plot the pixel values of column 1700 of the filtered HDR in Fig. 4.13; the signal is much smoother than Fig. 4.2a.

If there are coding artifacts (such as blocky artifacts) in the video, our algorithm can remove or at least reduce them. Pixels in regions with blocky artifacts usually have similar values, and our filter smoothes them out. Note that pixels in those regions can have different gradient directions, so the algorithms using directional features to detect and reduce artifacts may not work well. Our algorithm does not depend on the direction or gradient, so it is effective for blocky artifacts.

We compared our proposed filter against three debanding or dithering algorithms:



(a) inverse tone mapped HDR (b) filtered HDR

Figure 4.12: Results. Note that banding artifacts in (a) are more noticeable on a screen than on paper.

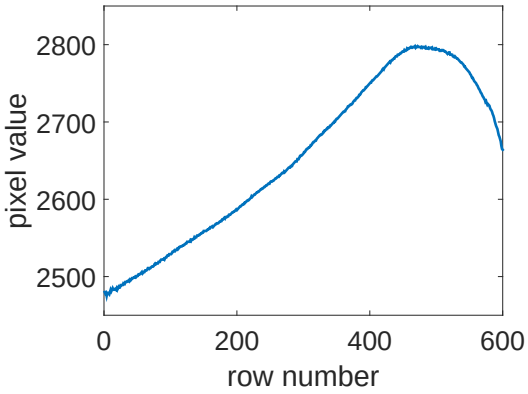


Figure 4.13: Pixel values of column 1700 of the filtered HDR where the input signal is Fig. 4.2a.

a) The method of Bhagavathy et al. [48]: banding is detected at each pixel by looking for pixels with value $b \pm 1$ in a neighborhood, where $x[m,n] = b$ (un-normalized value), and $x[m,n]$ is the central pixel of the neighborhood. Six neighborhoods ranging from 10×10 to 110×110 are tested. If at least one neighborhood satisfies the given criteria, the central pixel value is replaced by a weighted sum: $\tilde{y}[m,n] = g_{-1} \cdot (b-1) + g_0 \cdot b + g_1 \cdot (b+1)$, where the weights, g_{-1}, g_0 and g_1, depend on the ratio of pixels with values $b-1$, b, and $b+1$. This method works for 8-bit LDR images, but not for inverse tone mapped HDR images. The difference between two neighboring codewords in the inverse tone

mapped HDR is probably greater than 1, and depends on the iTMO. Therefore, we modified the method in [48] by applying the banding detection and computation of weights (Eqs. 1 and 2 in [48]) using the reconstructed 8-bit LDR image. Then we apply the weights to the inverse tone mapped HDR image (Eq. 7 in [48]): $y[m,n] = g_{-1} \cdot T(b-1) + g_0 \cdot T(b) + g_1 \cdot T(b+1)$, where $T(\cdot)$ is the inverse tone mapping function. Multi-scale neighborhoods have to be tested in order to obtain the best output. For each pixel, $3 \times 110 \times 110$ comparisons are conducted to check the number of pixels with values b, $b+1$ and $b-1$ in the neighborhood. For each size of neighborhood, three multiplications are used to compute the ratio of pixels with b, $b+1$ and $b-1$; three multiplications, two additions and one comparison are needed to compute the confidence score (Eq. 1 in [48]); and three comparisons are needed to check if the neighborhood satisfies the criteria (Eq. 2 in [48]). At most 5 comparisons are used to find the neighborhood with the highest confidence score. The weighted sum (Eqs. 5 and 7 in [48]) costs 4 multiplications and 2 additions. In total, this method requires 40 multiplications and 72672 additions for each pixel at the decoder. The computation is considerably more complex than our filter. If we want to move the computation of weights to the encoder, we must transmit the three weights of each pixel to the decoder, equivalent to sending three more images. The overhead would be very high.

b) Bilateral filter [58]: the filtering output is

$$y[m,n] = \frac{1}{W_p[m,n]} \sum_{i=-u}^{u} \sum_{j=-v}^{v} x[m+i,n+j] w_p[m,n,i,j], \tag{4.9}$$

where

$$
\begin{aligned}
w_d[i,j] &= e^{-\frac{i^2+j^2}{2\sigma_d^2}}, \\
w_r[m,n,i,j] &= e^{-\frac{(x[m,n]-x[m+i,n+j])^2}{2\sigma_r^2}}, \\
w_p[m,n,i,j] &= w_d[i,j] \cdot w_r[m,n,i,j], \\
W_p[m,n] &= \sum_{i=-u}^{u} \sum_{j=-v}^{v} w_p[m,n,i,j].
\end{aligned}
\tag{4.10}
$$

$x[m,n]$ is the inverse tone mapped HDR, and the output $y[m,n]$ is the filtered HDR. The weights depend on the distance and the difference to the central pixel, and are based on a Gaussian distribution. There are four parameters: the vertical span $2u+1$, the horizontal span $2v+1$, the spatial kernel sigma σ_d and the range kernel sigma σ_r. We set $u = v$, reducing the number of parameters to 3. We manually select the span and the two sigma values for each test image, ensuring that banding is removed from the image with the most details preserved by visual inspection. We set v to 14 for 5 out of the 8 test images, and set v to 24 for the other 3 test images. The bilateral filter is a dense filter. One may determine the parameters at the encoder by solving some optimization problem, but the decoder has to compute the weights. For a given set of parameters, look-up tables can be pre-built for each possible w_d and w_r. For each pixel, $(2v+1)^2$ additions are required to compute $x[m,n]-x[m+i,n-j]$, $(2v+1)^2$ multiplications for computing $w_p[m,n,i,j]$, $(2v+1)^2$ multiplications for computing $x[m+i,n+j]w_p[m,n,i,j]$, $(2v+1)^2$ additions for computing $W_p[m,n]$, $(2v+1)^2$ additions for computing the weighted sum, and one multiplication for dividing the weighted sum by $W_p[m,n]$. In total, the bilateral filter demands $2(2v+1)^2+1$ multiplications and $3(2v+1)^2$ additions for each pixel. For $v = 14$, it costs 1683 multiplications and 2523 additions at the decoder. The computation complexity is much higher than our proposed filter. If we compute the weights at the encoder and send the weights of each pixel to the decoder, the overhead would be

$(2v+1)^2$ images, which is infeasible.

c) Gaussian noise injection: we add zero-mean Gaussian noise to the reconstructed HDR. This is a simple method to cover banding and blocky artifacts in images. We select the standard deviation manually for each image so that banding becomes unnoticeable by visual inspection. Note that sometimes it is impossible to cover the banding even with extremely strong noise. The computation is lighter than our method. It costs only one addition for each pixel.

Besides the three filtering / dithering methods, we also compare the inverse tone mapped HDR without debanding.

4.4.1 Objective Comparisons

We compute the PSNR gains of the four debanding / dithering schemes over no debanding. The PSNR is measured in the banding regions which are the locations of the banding steps. The PSNR gains of the 8 inverse tone mapped HDR images are shown in Table 4.1. The average gains of our proposed filter, the method of Bhagavathy et al. and bilateral filter are almost the same. Note that the method of Bhagavathy et al. and the bilateral filter are dense filters, and are computational demanding. The Gaussian noise injection has significant PSNR loss in the banding regions.

The PSNR gains of the 4 video clips in the banding regions are shown in Table 4.2. Each test sequence is 10 - 15 sec at 24 fps. We do not include bilateral filtering because its computation is too intense, and we have to adjust the 3 parameters for each frame manually. The proposed method and the method of Bhagavathy et al. achieve almost the same gain. The noise injection has PSNR loss.

Table 4.1: PSNR gain (dB) over no debanding in the banding regions of test images

Image	Proposed	Bhagavathy et al.	Bilateral	Noise injection
1	3.76	5.06	3.16	-23.56
2	2.06	1.70	2.14	-13.68
3	0.62	0.54	0.44	-12.96
4	3.84	3.87	4.03	-19.46
5	1.99	1.95	1.95	-17.32
6	2.45	2.50	2.40	-20.86
7	2.38	2.62	2.52	-20.65
8	3.35	3.84	4.01	-18.49
Average	2.56	2.76	2.58	-18.37

Table 4.2: PSNR gain (dB) over no debanding in banding regions of test sequences

Sequence	Proposed	Bhagavathy et al.	Noise injection
1	3.09	3.13	-19.96
2	1.20	1.17	-17.26
3	0.31	0.98	-23.84
4	0.53	0.41	-9.95
Average	1.29	1.42	-17.75

4.4.2 Subjective Test

We also evaluate the performance of our debanding filter by a subjective test with 11 observers. The subjective test included two sessions. In the first session, subjects compared images in pairs: one image processed using our proposed debanding filter, and the other is from the method of Bhagavathy et al., the bilateral filter, Gaussian noise injection, or no debanding. The randomized images were labeled A and B. Subjects were given 5 options: “A is much better than B”, “A is slightly better than B”, “A is the same as B”, “A is slightly worse than B”, and “A is much worse than B”. Subjects were also asked to select the reasons why they prefer one to the other one. The possible reasons were: less banding, more details, less noise or other artifacts.

The second session involved video quality evaluation. Subjects rated the quality of each video sequence individually on a 5-point scale: “5 - excellent”, “4 - good”, “3 -

fair", "2 - poor", and "1 - bad". Four schemes were included: no debanding, our proposed method, the method of Bhagavathy et al., and Gaussian noise injection. We again asked subjects to select reasons for their ratings if they rated the quality below good. The possible reasons were: banding, loss of details, too noisy or other spatial artifacts, and temporal flickering.

Before the formal test, we ran a training session to ensure the subjects were familiar with the procedure and the rating system. The whole experiment took about one hour with several breaks. The visual testing was conducted on a Dolby Pulsar 4,000 nits HDR monitor.

Image comparison

The difference mean opinion score (DMOS) is computed between our proposed filter and the other schemes. Positive (negative) numbers mean our proposed debanding filter works better (worse) than the other scheme. We plot the DMOS and the 95% confidence intervals (CIs) in Fig. 4.14. The CIs for the comparison against no debanding are above zero for 7 out of 8 images, which shows our proposed filter improves the quality effectively.

For the comparison against the method of Bhagavathy et al., our proposed method shows advantage for two images. For image 2, eight subjects prefer our proposed filter to the method of Bhagavathy et al. because the output of our filter has less banding. Two subjects favor the method of Bhagavathy et al. due to detail preservation. One subject thinks there is no difference. For image 3, ten out of eleven subjects prefer ours due to less banding. The other one prefers the method of Bhagavathy et al. slightly because it shows more details. We plot the pixel intensities of one row of image 3 in Fig. 4.16. The output of our filter is smoother than the output of the method of Bhagavathy et al. Some pixels in the output of Bhagavathy et al. (Fig. 4.16b) are not smoothed, because

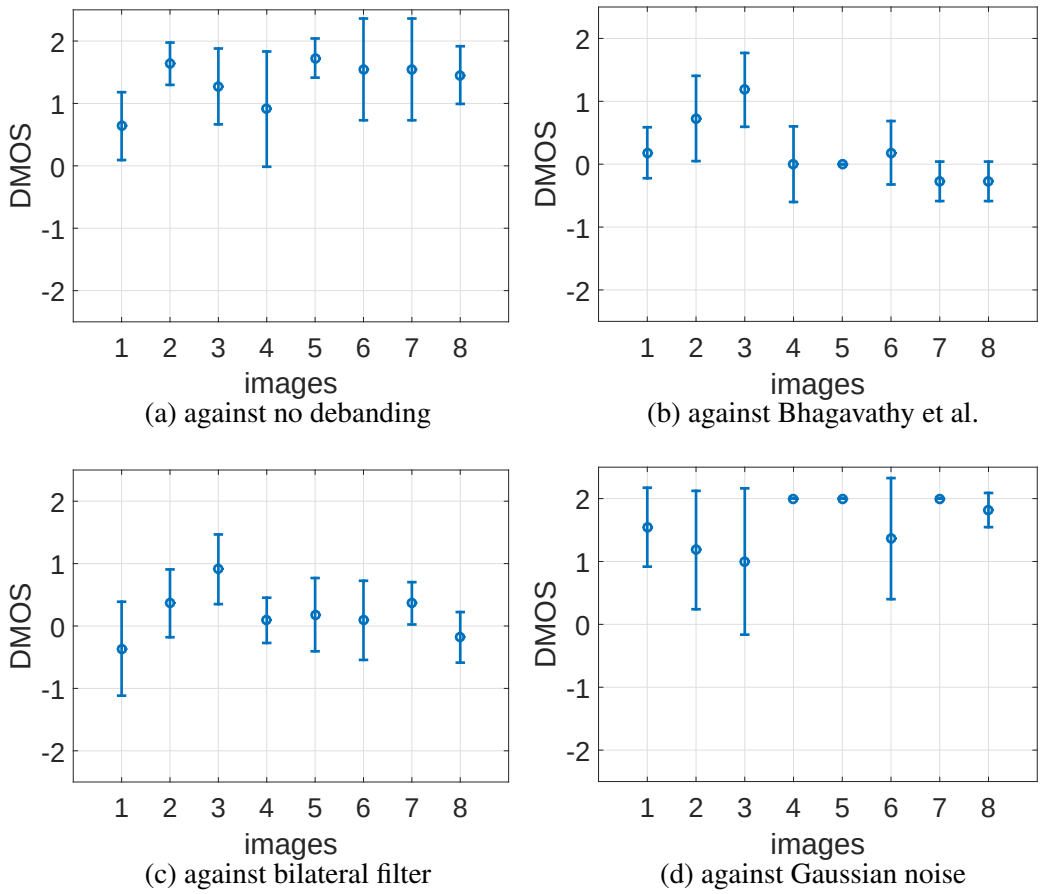


Figure 4.14: 95% confidence intervals of image comparison DMOS of proposed scheme vs. other schemes

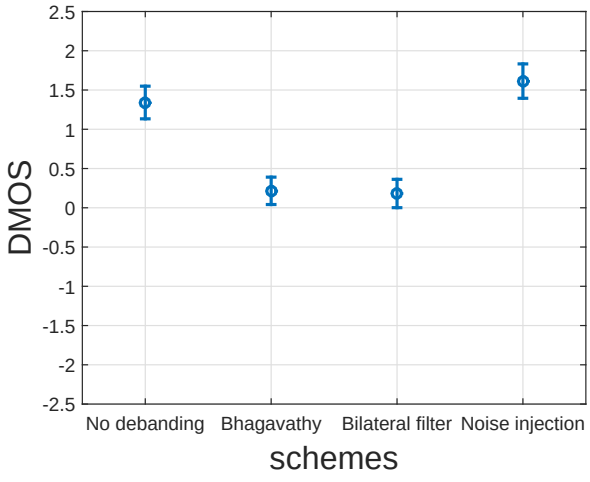


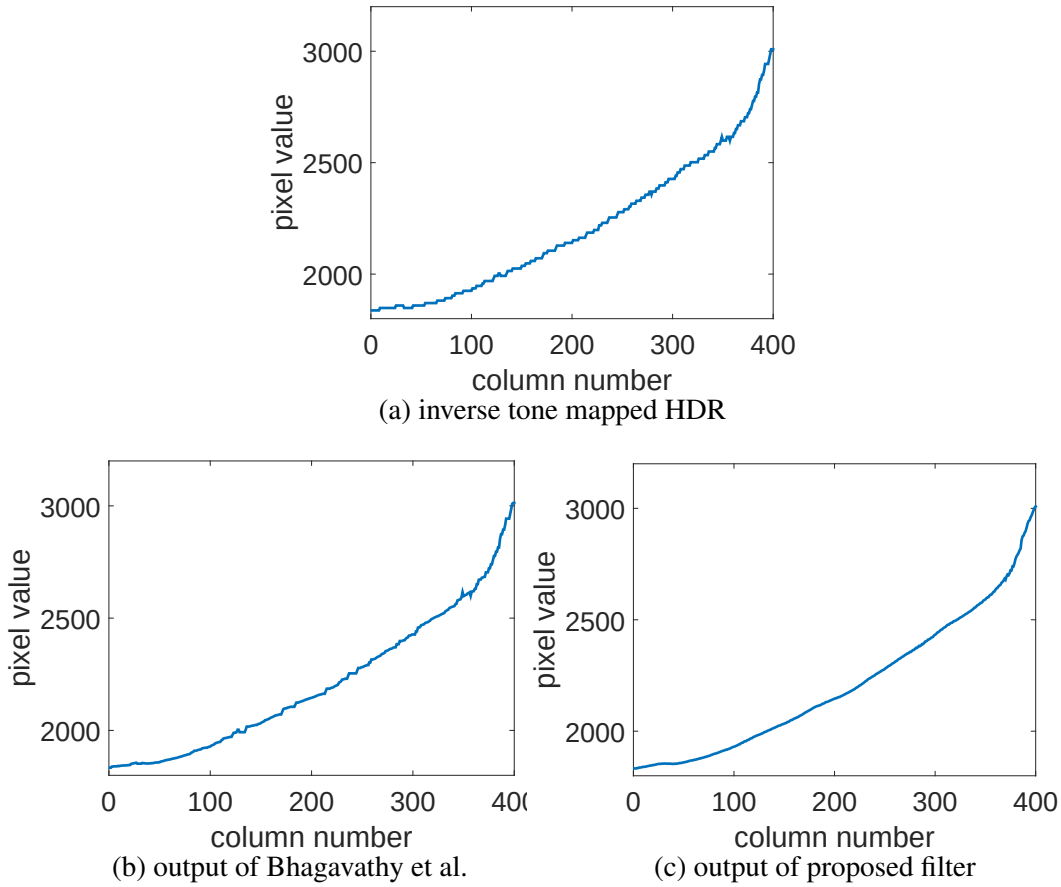
Figure 4.15: 95% confidence intervals of DMOS of proposed scheme vs. other schemes for all test images

the reconstructed 8-bit LDR pixel values in the neighborhood are two codewords from the value of the central pixel. Therefore, no neighborhood satisfies the given criteria in [48]. In order to make the method of Bhagavathy et al. work for this case, one has to modify the criteria in [48]. The weighting would become more complicated. For the other six images, we cannot reject the null hypothesis that the two methods achieve the same performance.

Our proposed filter is slightly better than the bilateral filter for two images. The reason is that our proposed filter preserves more details. That is because the range kernel sigma of the bilateral filter, σ_r, is the same for all pixels, regardless of the iTMO. The range kernel sigma has to be large enough to smooth out banding over the entire image, whereas some areas which need smaller σ_r are blurred. Making σ_r a function of the iTMO is a possible way to improve the bilateral filter, but it requires further study and is not in this work's scope. The bilateral filter achieves the same performance as our proposed filter for the other 6 images.

Our proposed filter outperforms Gaussian noise injection for 7 out of 8 images. Most subjects think the latter is too noisy. For the other image, subjects have different preferences, so we cannot reject the possibility that the two schemes achieve the same debanding effect. According to the averaged DMOS, the advantage of our filter over Gaussian noise injection is even larger than the advantage over no debanding.

When we pool the DMOS of all the test images, our proposed filter outperforms all the other schemes. In Fig. 4.15, CIs are computed using the scores of all the images. The CIs are all above zero. Our proposed filter has significant advantage over no debanding and Gaussian noise injection, where the average DMOS is 1.34 and 1.61. Our proposed filter also shows a slight advantage over the method of Bhagavathy et al. and bilateral filtering by 0.22 and 0.18.



(a) inverse tone mapped HDR

(b) output of Bhagavathy et al.

(c) output of proposed filter

Figure 4.16: Pixel values of row 435 of image 3

Video quality evaluation

We compute the mean opinion score (MOS) of each video sequence (Table 4.3). Our proposed filter and the method of Bhagavathy et al. yield good quality on the average. No debanding and Gaussian noise injection are poor in quality.

We compute the DMOS of our proposed filter versus other schemes and plot the 95% CIs in Fig. 4.17. It is clear that our proposed filter performs much better than no debanding and Gaussian noise injection for all the test sequences. We cannot reject the null hypothesis that our proposed method performs the same as the method of Bhagavathy et al. for 3 out of 4 sequences. For Sequence 2, our proposed method has a small advantage over the method of Bhagavathy et al.

Table 4.3: MOS of video sequences

Sequence	No deband	Proposed	Bhagavathy	Noise injection
1	2.91	4.09	3.91	2.18
2	2.36	4.36	3.82	1.55
3	2	4.27	4	1.82
4	1.91	4.18	3.73	1.82
Average	2.30	4.23	3.86	1.84

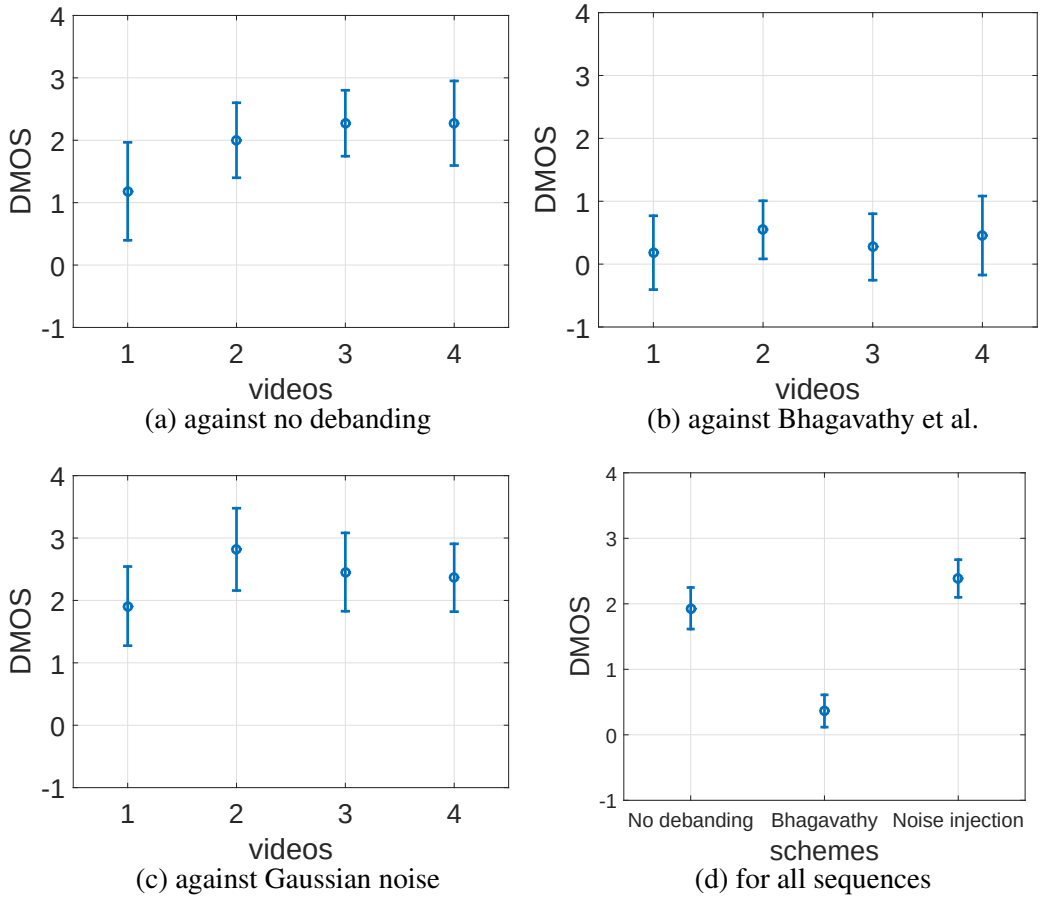
Table 4.4: Percentage of time when artifacts are reported

Schemes	No deband	Proposed	Bhagavathy	Noise injection
Banding	95%	18%	43%	75%
Detail loss	2%	7%	9%	0
Noise	0	0	0	95%
Flickering	5%	5%	9%	14%

When we pool the DMOS of all the test sequences, our proposed filter wins over all the other schemes. All the CIs of Fig. 4.17d are above zero. We even have an advantage over the method of Bhagavathy et al.

During the rating of no debanding, banding artifacts were reported 95% of the time (Table 4.4). This is reduced to 18% when our proposed filter is rated, while banding is reported in the output of Bhagavathy et al. 43% of the time. That means our proposed filter is more effective at removing banding. The detail preservation of the two methods is almost the same.

When rating Gaussian noise injection, subjects disliked the noise, and reported banding 75% of the time. In the image comparison between our proposed filter and Gaussian noise injection, banding is reported in the noise injection scheme only 14% of the time. This is because banding is not masked well in moving pictures, especially when banding is moving. The high frequency noise does not make banding in low temporal frequency invisible.



(a) against no debanding (b) against Bhagavathy et al.

(c) against Gaussian noise (d) for all sequences

Figure 4.17: 95% confidence intervals of video DMOS of proposed scheme vs. other schemes

4.5 Summary

In this chapter, we propose an edge-aware selective sparse filter to remove banding artifacts and reduce coding artifacts in inverse tone mapped HDR videos. The main contributions are summarized as follows:

1. The filter combines non-smooth area detection and filtering. No banding map or filtering map is required to store at, or transmit to, the decoder. The filter can be implemented and executed in hardware efficiently at the decoder.

2. The inverse tone mapping function is considered when performing the non-smooth area detection, which helps detail preservation in the entire image.

3. The filter uses 7 taps to detect non-smooth areas and only 5 taps to do the filtering. It prevents introducing more ringing artifacts, and further helps to preserve edges and details.

4. The parameters of the filter, the span and the threshold factor α in the decision process, are selected by minimizing a perceptual distortion metric at the encoder. The parameters can be sent to the decoder as metadata.

5. Significant PSNR gain at the regions of artifacts is obtained after filtering. Subjective tests show our proposed filter has significant advantage over Gaussian noise injection and no debanding. It performs at least as well as the method of Bhagavathy et al. and bilateral filtering, but requires much lighter computation. The proposed filter outperforms the method of Bhagavathy et al. and bilateral filter for some contents.

Acknowledgment

Chapter 4, in part, is a reprint of material as it appears in Q. Song, G.-M. Su, and P. C. Cosman, "Efficient debanding filtering for inverse tone mapped high dynamic range videos", submitted to *IEEE Transactions on Image Processing*, and Q. Song, G.-M. Su, and P. C. Cosman, "Hardware-efficient debanding and visual enhancement filter for inverse tone mapped high dynamic range images and videos", *International Conference on Image Processing*, pp. 3299-3303, Sep. 2016. The dissertation author was the primary author and the co-author Dr. Su directed and supervised the research which forms the basis for Chapter 4. The co-author Prof. Cosman also supervised this work.

Chapter 5

Packet Loss Visibility of 2D+Depth Compressed Stereo 3D Video

In the previous chapters, we discuss the perceptual quality enhancement and preservation of 2D videos. In this chapter, we investigate 2D+depth stereoscopic 3D LDR videos. We conduct a subjective test on the visibility of fixed-sized packet losses in 3D videos. We construct a model to predict the loss visibility (i.e., the importance) of packets using features extracted from the video. The model can be used for unequal error protection during transmission. Strong protection can be applied to packets with high visibility by allocating more forward error correction to them.

In the following, we introduce the 2D+depth format of 3D video compression in Sec. 5.1. The subjective test is described in Sec. 5.2, and the prediction model is explained in Sec. 5.3. Sec. 5.4 summarizes the chapter.

Figure 5.1: A left color view and its depth map

5.1 2D+Depth Coding Format

For stereoscopic 3D video, the 2D+depth format consists of the left color view and its depth map. The depth map is a 2D representation of the 3D surface. It includes the distance of objects in the scene from the camera but no information of texture. The depth map is a grayscale image, thus can be compressed in YUV 4:0:0 format (Fig. 5.1). It can be generated from the original left and right color view. In recent years, depth estimation has been extensively explored, and many of the algorithms are evaluated by the Middlebury Stereo Benchmark [59]. In our observer experiment, we employ the widely used Min-Cut algorithm [60] which is also used in MPEG Depth Estimation Reference Software [61]. The left color view and the depth map can be separately or jointly encoded.

At the decoder, the right view is synthesized from the decompressed left color view and depth map. If the two views are well rectified and parallel, the right view can be synthesized efficiently without a z-buffer [62]. The columns of the left image are warped from left to right image borders based on the 3D structure built from the depth information. The major problem is disocclusion. Some areas occluded in the left view can be visible in the right view. This results in holes in the synthesized right view. One solution is to fill the holes by spatial interpolation using neighboring pixels. Another one is preprocessing the depth map with a Gaussian filter so that much smaller disocclusion

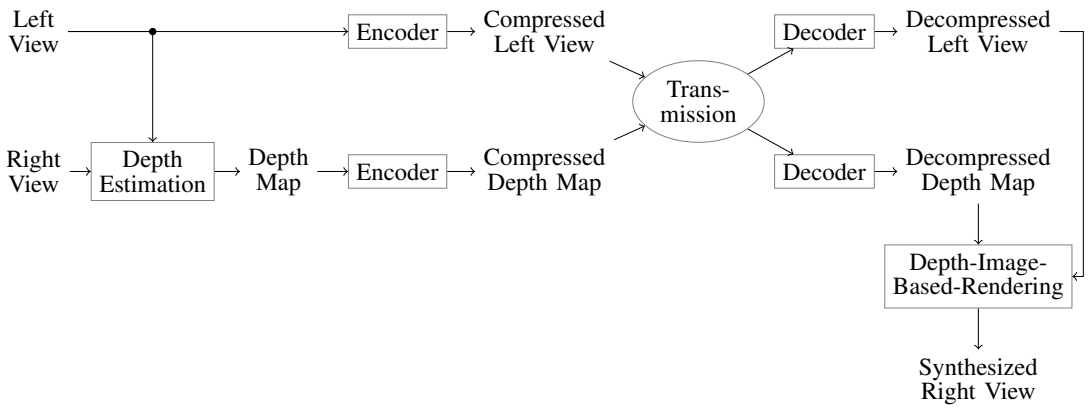


Figure 5.2: 2D+depth block diagram

areas would appear in the right view [63]. Based on [64], the Gaussian filtering method yields the best visual quality. It only incurs a small geometric distortion but no visible flickers along the object edges. Therefore, in our observer experiment we also apply a 27×27 Gaussian filter with $\sigma = 9$ to preprocess the depth map before 3D warping. If disocclusion still remains after the filtering, the holes would be diminished to a very small area, and then we use the spatial interpolation method proposed in [65], which has similar performance to inpainting [66] but works more efficiently.

5.2 Human Observer Experiment

5.2.1 Motivation

As stated in Chapter 1, video packets can be corrupted during transmission, and packet losses can have different visual impacts. Hewage et al. found that the overall video quality is affected by both color view and depth packet losses, but prioritizing color 2D video packets can vastly improve the overall video quality [67][68]. Pinto et al. mentioned that losses in videos with low disparity is less annoying than in ones with high disparity [69]. However, they only evaluate the overall image quality and depth

perception of the entire video sequence via PSNR or MOS (Mean Opinion Score). It remains unclear if an arbitrary color view packet has greater impact than any depth packet. For example, whether a P frame in the color view video is more important than an I frame in the depth map video was not addressed. It is also unclear if a packet located in a low disparity region always causes less damage than a packet in a high disparity region. One may ask if other factors, such as the spatial location of the packet, can also affect the perceived quality. Simple objective metrics may not be satisfying to represent the complex 3D attributes. Therefore, we conduct a human observer experiment to measure the human perception of different types of packet loss. The video is encoded in 2D+depth format and packetized into fixed-sized packets.

5.2.2 Setup of the Experiment

We conducted a human observer experiment in which the viewers were shown 3D videos with impairments caused by packet losses. The viewers were asked to press the space bar once they saw a glitch. To allow the viewers to have enough responding time, we insert at most one loss in every 4 seconds. The loss occurred in the first 3.2 seconds in each 4-sec interval, and the last 0.8 seconds would allow any error propagation to stop. The viewer was considered as having seen the loss if he/she responded within 1 second after the loss.

We encoded the left color view (denoted as *color* or *color video* below) and the depth map (denoted as *depth* or *depth video* below) separately, as we want to compare the impacts of losses in color and in depth. The encoder is H.264 JM 18.1 [70]. The color video is in YUV 4:2:0 format and the depth video is in 4:0:0 format. Quantization Parameter (QP) values of 26, 31, 36 and 41 for both color and depth are suggested in [71]. It was found that increasing the bit-rate of depth can improve the quality of the synthesized right color view significantly [62]. Thus, we fix QP to 26 for both color and

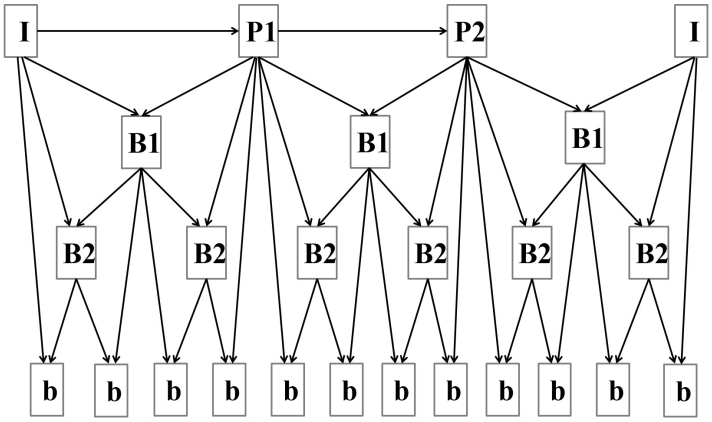


Figure 5.3: Hierarchical GOP structure

depth video. The test video is 21'20" long. The color video is HD (1920×1080), and has 30 frames per second. The depth video is downsampled by 2 in the horizontal and vertical direction, so the size of each depth frame is a quarter of a color frame, which is also suggested in [71]. The deblocking was turned off when depth was encoded. We use the hierarchical GOP structure and insert intra frames every 24 frames (0.8 sec per I frame, Fig. 5.3). There are 6 types of frames in the GOP: one type of I frame, two types of P frames and 3 types of B frames, classified by their time duration. The time duration, or the maximal length of error propagation, is defined as the maximal number of frames affected by the error in one frame. For example, any loss in a P1 frame would affect itself, the next P2 frame and 21 B frames. The time duration of each type of frame is given in Table 5.1. The video bitstream is divided into fixed-sized packets of 1316 bytes (equal to seven MPEG packets of 188 bytes in length), as recommended in [72]. Each packet includes at most one frame and would not include any information of the next frame. A packet would not split a macroblock either. So some packets could be less than 1316 bytes.

The decoder is JM 16.2. To conceal losses in color and depth I frames, we use spatial interpolation by taking the sum of weighted neighboring available pixels. To conceal losses in color P or B frames, we use motion-compensated error concealment

Table 5.1: Maximal Number of Frames Affected

Frame Type	Time Duration
I	31
P1	23
P2	15
B1	7
B2	3
b	1

(MCEC). The motion vectors of neighboring available (correctly decoded or concealed) macroblocks are extracted. The motion vector that minimizes the boundary matching error [73] is taken to conceal the corrupted macroblock. If the neighboring macroblock is sub-partitioned, only the motion vectors of the blocks adjacent to the macroblock to-be-concealed are considered as candidates. For example, corrupted macroblocks are shown in dark gray in Fig. 5.4. To conceal MB #1, motion vectors of the blocks in light gray above and below MB #1 are considered as candidates when those macroblocks are sub-partitioned. Only the motion vectors of correctly decoded macroblocks are considered if they are available. The motion vectors of concealed macroblocks are considered only when none of the neighboring macroblocks is correctly decoded (the macroblock to be concealed is surrounded by other corrupted macroblocks). In Fig. 5.4, only the motion vectors of the blocks below MB #2 are candidates for MCEC. Though the macroblocks above MB #2 are concealed first, their motion vectors are considered unreliable so we do not use them. If all the neighboring macroblocks are intra-coded or the whole frame is lost, then no motion vector is available. In that case, we set the motion vector to zero, i.e., copying the co-located macroblock from the reference frame.

For losses in depth P or B frames, we conceal each macroblock by setting its motion vector as half of the average of the motion vectors extracted from the co-located macroblocks in the corresponding color frame, due to the high correlation between color

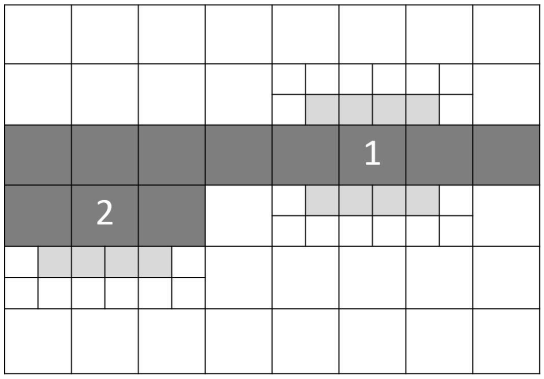
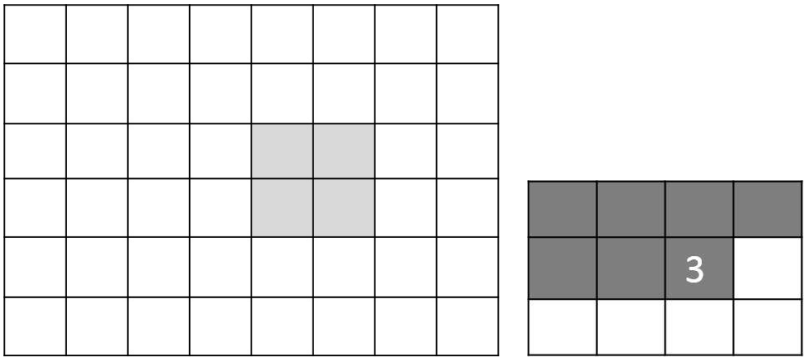


Figure 5.4: Error concealment for color frame.



(a) Color frame (b) Depth frame

Figure 5.5: Error concealment for depth frame.

and depth video. In our experiment, the co-located macroblocks in the color frame are always available (though their motion vectors may not exist), because there is at most one packet loss in every 4-sec interval, so if the loss is in a depth frame, then the corresponding color frame is intact. Since the depth maps are downsampled by 2 in each direction, the motion in the color frame can be twice the motion in the depth, and one macroblock in a depth frame corresponds to 4 macroblocks in the color frame. In Fig. 5.5, macroblocks in light gray shade in the color frame are extracted to conceal MB #3 in the depth frame. The efficiency of this method is shown in [74]. If all of the co-located color macroblocks are intra-coded, we simply set the motion vector of the corrupted depth macroblock to zero.

We generated 5 versions of the lossy video. Each version includes 300 packet losses. These losses are divided equally and randomly among each type (color and depth, each has 6 types). Each version of the lossy video was evaluated by 12 viewers. All of the viewers have normal or corrected-to-normal vision, and have good stereo vision (tested by the stereo fly test). Before the experiment, a 3-min pilot training video was shown so that the viewers could get a sense of the artifacts they were going to see. The lossy videos also include some intervals without any loss so that we can measure the false positive rate caused by factors other than the packet losses, such as view synthesis artifacts.

5.2.3 Experimental Results

We define the visibility score of each packet as the number of viewers who saw its loss divided by the total number of viewers who assessed that version of lossy video. Fig. 5.6 shows the mean visibility score of each type of loss. The visibility of losses in color frames is generally higher than losses in depth. One reason is that a color packet loss would affect the left color view itself and the right color view rendered from it. However, if a depth packet is lost, only the right color view would be affected. Another reason is that color packet losses usually cause blocky artifacts, which are probably more likely to be seen than the geometric distortion caused by depth losses.

Among the color packet losses, losses in P frames are the most visible. One might have expected that losses in I frames should be the most damaging since they have the longest error propagation, and this in fact has been true in the case where packets hold a fixed number of macroblocks. However, as we fix the size in bytes of each packet, the spatial area affected by a packet loss in an I frame is usually less than the area affected in a P frame which is less than the area affected in a B frame. Table 5.2 shows the average number of packets included in each type of frame. One packet in a color I frame covers

Table 5.2: Average number of packets included in a frame

Video	I frame	P frame	B frame
Color	50.8	39.3	20.4
Depth	2.3	2.0	1.5

on average 1.97% of the spatial area of a frame, while a packet in a color P frame covers 2.55% area. So under the interaction of time duration and spatial area affected, losses in P frames have the highest visibility score. This is consistent with the previous work [3].

For the depth packet losses, it turns out that losses in I frames are the most prominent. While a depth I frame packet does cover slightly less spatial area than a depth P frame packet, the visibility scores for depth packets do not follow the same trend as color packets because the error concealment for depth is quite different. Losses in depth P and B frames can be concealed better than losses in I frames. Motion in the color frame and the depth frame is highly correlated. Besides, depth frames include very little texture, so the residual energy after motion compensation is usually small. Therefore, copying the motion vectors of the corresponding color frames is very helpful to recover the corrupted macroblocks. There are no motion vectors in I frames, and the spatial interpolation often yields an unsatisfying result when the corrupted area is large.

In the experiment, there are twenty 4-sec intervals without any loss in each version of the video. We collected the viewers' responses in those intervals to measure the false positive rate. False positive responses may be due to compression artifacts, view synthesis artifacts, or just inattention. The false positive rate is 4.17%, which is well below the mean visibility score of losses in all the color frame types and in depth I and P frames. However, the mean visibility scores of packet losses in depth B1, B2 and b frames are 0.0560, 0.0787 and 0.0333 respectively, which are close to the false positive rate. This suggests that some or most of the responses counted for these losses may not actually be due to the losses. It would be wrong to conclude however that all depth B

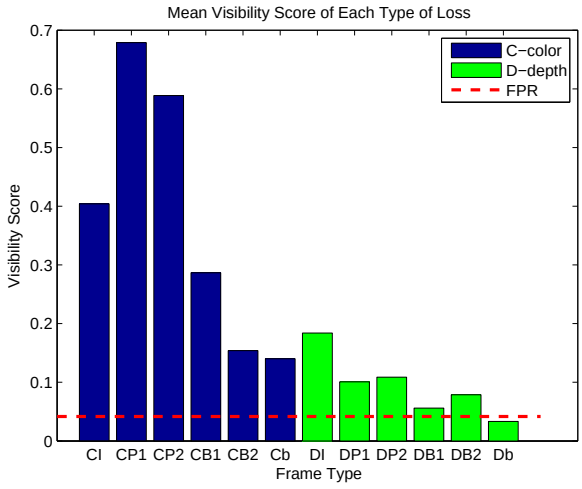


Figure 5.6: Mean visibility score of each type of loss. The dash line shows the false positive rate, which is 0.0417.

frame losses can be assumed to be unimportant visually, because different packets of the same type sometimes have very different visibility scores. For example, some losses in depth b frames have visibility score as high as 0.75, though most losses in that frame type were not perceived by any viewers. The mean visibility score of losses in color P1 frames is 0.6787 and 30.4% of that type of losses were seen by all the viewers, but some other packets of that type have zero visibility score. So the mean value may not well represent the visibility score of each loss. Therefore, we aim to investigate the features of each packet and use them to predict the visibility score.

5.3 Visibility Model

Since the main use of the predictions of the visibility score would be for unequal error protection, we want to make the prediction at the encoder side. That means we have access to the original video, the compressed bitstream and the reconstructed video at the encoder. To predict the visibility score, we extract features from the videos and bitstream, and then utilize those features to build a visibility model. We first describe the features in

Table 5.3: Content Independent Features

Feature	Abbreviation	Description
IsColor	IsColor	Packet is in color frame
Time Duration	TMDR	Maximal number of frames affected
Deviation from Border	DevFromBorder	floor(N/2) − |Height-floor(N/2)|, N is number of rows of macroblocks
Frame Type	IsCIframe	Packet is in color I frame
	IsCPframe	Packet is in color P frame
	IsCBframe	Packet is in color B frame
	IsDIframe	Packet is in depth I frame
	IsDPframe	Packet is in depth P frame
	IsDBframe	Packet is in depth B frame

this section, then explain the modeling approach and the results.

5.3.1 Feature Extraction

The extracted features are grouped into two categories: content independent features and content dependent features. The feature abbreviations and brief descriptions are given in Table 5.3 and 5.4.

Content independent features, such as the frame type determined by the GOP structure and the spatial location of the packet, do not depend on the content of the video. The following features are considered:

1. IsColor: a boolean factor which is set to 1 if the packet is in a color frame, and is set to 0 if it is in a depth frame.

2. Time Duration (TMDR): the maximal number of frames affected by the loss, which is completely determined by the type of frame that includes the packet. (Table 5.1)

3. Deviation from Border (DevFromBorder) = floor(N/2) − |Height-floor(N/2)|, where Height is the vertical location of the packet center, N is the number of rows of macroblocks in one frame. N = 68 in this experiment since we use HD video.

4. IsCIframe, IsCPframe, IsCBframe, IsDIframe, IsDPframe and IsDBframe are boolean factors denoting the frame type. IsCPframe means the packet is in a color P frame. We do not specify P1 and P2 so that the prediction model can be used for other GOP structures and I frame periods.

Table 5.4: Content Dependent Features

Feature	Abbreviation	Description
Number of MBs	NumMB	Number of macroblocks (MBs) in packet if IsColor = 1, 4 times number of MBs in packet if IsColor = 0
Packet Size	PktSize	Number of bytes in packet
Number of MBs Coded in a Certain Mode	CNumIntra	Number of color MBs in affected area which are intra coded
	CNumInter	As above, inter coded
	CNum(Skip/Direct)	As above, skip or direct coded
	DNumIntra	Number of depth MBs in affected area which are intra coded
	DNumInter	As above, inter coded
	DNum(Skip/Direct)	As above, skip or direct coded
Ratio of MBs Coded in a Certain Mode	CIntraRatio	CNumIntra / NumMB
	CInterRatio	CNumInter / NumMB
	C(Skip/Direct)Ratio	CNum(Skip/Direct) / NumMB
	DIntraRatio	DNumIntra / (NumMB / 4)
	DInterRatio	DNumInter / (NumMB / 4)
	D(Skip/Direct)Ratio	DNum(Skip/Direct) / (NumMB / 4)

Table 5.4: Content Dependent Features, Continued

Feature	Abbreviation	Description
Max Sub-partitions	CMaxInterparts	Maximal sub-partitions in affected color MBs
	DMaxInterparts	Maximal sub-partitions in affected depth MBs
Motion Vector	CMaxMotX, CMeanMotX, CVarMotX	Maximum of absolute value, mean and variance of horizontal motion vectors(MVs) of affected color MBs
	CMaxMotY, CMeanMotY, CVarMotY	Maximum of absolute value, mean and variance of vertical MVs of affected color MBs
	CMaxMotM, CMeanMotM, CVarMotM	Maximum, mean and variance of MV magnitude of affected color MBs
	CMaxMotA, CMeanMotA, CVarMotA	Maximum of absolute value, mean and variance of motion direction of affected color MBs
	DMaxMotX, DMeanMotX, DVarMotX	Maximum of absolute value, mean and variance of horizontal MVs of affected depth MBs
	DMaxMotY, DMeanMotY, DVarMotY	Maximum of absolute value, mean and variance of vertical MVs of affected depth MBs
	DMaxMotM, DMeanMotM, DVarMotM	Maximum, mean and variance of MV magnitude of affected depth MBs

Table 5.4: Content Dependent Features, Continued

Feature	Abbreviation	Description
	DMaxMotA, DMeanMotA, DVarMotA	Maximum of absolute value, mean and variance of motion direction of affected depth MBs
Residual Energy	CMaxRSENGY	Maximum of residual energy of affected color MBs after motion compensation
	DVarRSENGY	Variance of residual energy of affected depth MBs after motion compensation
MSE	MaxMSE, MeanMSE, VarMSE	Maximum, mean and variance of MSE per MB
SSIM	MinSSIM, MeanSSIM, VarSSIM	Minimum, mean and variance of SSIM per MB
Foreground MBs	FGNum	Number of foreground MBs in packet
	FGRatio	FGNum / NumMB

Content dependent features are those related to the content of the video, such as motion complexity. We extract some of them from both color and depth videos. If the lost packet is in a color frame, the features of the macroblocks contained in the packet and features of the co-located depth macroblocks are extracted. Likewise for a loss in a depth frame, we extract information from itself and the co-located color macroblocks.

1. Number of macroblocks affected by the packet loss (NumMB). It denotes the area in the frame affected by the loss. For packets in color frames, NumMB is the number of macroblocks in the packet. For packets in depth, NumMB equals 4 times the number

of macroblocks in the lost packet, since one macroblock in the depth map corresponds to 4 macroblocks in the right color view synthesized from it. This feature relates to the frame type, the spatial correlation and the motion complexity. For example, in a P or B frame, if the motion is complicated, the residual energy after motion compensation would be high, then more bits would be allocated to code the macroblocks and a packet would include fewer macroblocks than would one which contains macroblocks from a low motion frame.

2. Packet Size (PktSize). Since each packet contains at most one frame, some packets could be less than 1316 bytes. Most of the values of PktSize are around 1316 as we fix the length of the packet. In the following two scenarios, the packet can be much less than 1316 bytes: (1) the whole frame is included in one packet and (2) the packet is the last one in that frame. So this feature may relate to spatial location and motion complexity. In the videos we use in this experiment, only a small number of color B frames, and some depth P and depth B frames are packetized into one packet, as the videos are HDTV.

3. Number of macroblocks coded in intra, inter and skip/direct mode (CNumIntra, CNumInter, CNum(Skip/Direct), DNumIntra, DNumInter and DNum(Skip/Direct)). Once we get the location of a lost color packet, we extract the mode of macroblocks in the lost color packet and the mode of co-located macroblocks in the depth frame. Similarly, for a depth packet loss, we extract the mode of macroblocks in the packet and the mode of the co-located color macroblocks. CNumIntra denotes the number of macroblocks located in the affected area in the color frame which are coded in intra mode; and DNum(Skip/Direct) denotes the number of macroblocks in the affected area in the depth frame which are coded in skip or direct mode.

4. Ratio of macroblocks coded in intra, inter and skip/direct mode (CIntraRatio, CInter-Ratio, C(Skip/Direct)Ratio, DIntraRatio, DInterRatio and D(Skip/Direct)Ratio) is the

number of macroblocks coded in that mode divided by the number of macroblocks in the affected area. These features relate to the motion of the affected area. For example, if the packet is in a P frame and the IntraRatio is very high, that probably means the motion is complicated and the error could be hard to conceal.

5. CMaxInterparts and DMaxInterparts are the maximal number of sub-partitions in the color and depth macroblocks lying in the affected area, respectively. If the MaxInterparts is large, it probably also implies complicated motion.

6. MotX and MotY are the motion vector components in the horizontal and vertical directions. MotM is the magnitude of the motion vector ($MotM = \sqrt{MotX^2 + MotY^2}$). MotA is the direction of the motion ($MotA = \arctan(MotY/MotX)$). We compute the maximum of the absolute value, mean, and variance of MotX, MotY, MotM and MotA of both color and depth macroblocks in the affected area. If all the macroblocks in the area are coded in intra mode, then all those values are set to 0. We use DMeanMotM to denote the mean value of the motion vector magnitude of the affected depth macroblocks.

7. RSENGY is the residual energy per pixel after motion compensation of the macroblock. We compute the maximum of the residual energy of color macroblocks (CMaxRSENGY) and the variance of the residual energy of depth macroblocks (DVarRSE NGY) in the affected area. The residual energy of depth macroblocks is usually small as they include little texture. But it can have a large value when the object is moving in the z direction. Its variance over the affected macroblocks can also relate to the motion complexity.

8. For each packet loss, MSE and SSIM (Structural Similarity Index) [9] per macroblock are computed between the compressed (error-free) video and the decompressed video with that one packet loss. We do not compute those values between the original raw video and the decompressed video with the packet loss because we are only interested

in the quality degradation caused by the packet loss, not by compression artifacts. We compute only the initial error caused by the packet loss within the frame where the loss occurs, instead of computing cumulative error over all the frames affected. This helps to reduce computational complexity. We then take the maximum, mean and variance of MSE per MB (MaxMSE, MeanMSE, VarMSE), and minimum, mean and variance of SSIM per MB (MinSSIM, MeanSSIM, VarSSIM). A large value of MSE and a small value of SSIM indicate large degradation in quality.

9. Viewers are usually attracted by foreground objects which may have different motion from the background. The cameras may also focus on those objects so the background may be blurry. So errors in the background can usually be concealed better than errors in the foreground. If most of the affected area is background, it may be less likely for the viewers to notice the packet loss. With depth maps, it is easy to extract foreground pixels from the frame. Pixels with depth deeper than some threshold are considered as background. To find a good threshold, we first plot the histogram of the depth values in that frame. Then we pick the minimum between the two non-neighboring highest bins as the threshold. In each macroblock, if over half of the pixels are foreground, we consider it as a foreground macroblock. FGNum is the number of foreground macroblocks in the packet. FGRatio is the portion of foreground macroblocks in the affected area, which is equal to FGNum divided by the total number of macroblocks in the packet.

5.3.2 Modeling Approach

We employ the generalized linear model (GLM) with logit as the link function for binomial distribution to build the prediction model. The inputs of the model are the features of a packet, and the output is the prediction of the packet's visibility score. The

model is

$$\log(\frac{p}{1-p}) = \gamma + \sum_{j=1}^{K} x_j \beta_j$$

where p is the visibility score, x_j is a feature, β_j is its coefficient, and γ is a constant term.

The whole dataset includes 1500 samples. We use 5-fold cross validation to select the most important features and prevent overfitting. The whole data is divided into 5 partitions, 4 of which are used to train the model and the one left out is used to test the performance. The procedure is repeated 5 times. Different partitions are used as test data each time. The optimal feature is selected in each step to minimize the mean squared error (MSE) between the predicted visibility score and the ground truth. A fixed set of features is used to train only one model.

5.3.3 Performance

We use mean squared error (MSE) and correlation coefficient to measure the performance of the model. We compute the two metrics between the prediction and the ground truth via 5-fold cross validation. Fig. 5.7 shows the performance vs. the number of features added into the model. The correlation coefficient reaches 0.75 when 30 features are added into the model, 0.72 with 20 features, 0.70 with 10 features and 0.67 with only 5 features.

Table 5.5 shows the ten most important features in the prediction model, where $\times$ means multiplication of the two single features. IsColor plays a key role since color packet losses are generally more visible than depth packet losses. Three out of the top ten features relate to IsColor, and their coefficients all have positive signs. The most important one is IsColor $\times$ CIntraRatio. It indicates that if the packet is in a color frame, and more macroblocks are coded in intra mode, the packet is more likely to be seen. That is because those corrupted macroblocks are not likely to be concealed well.

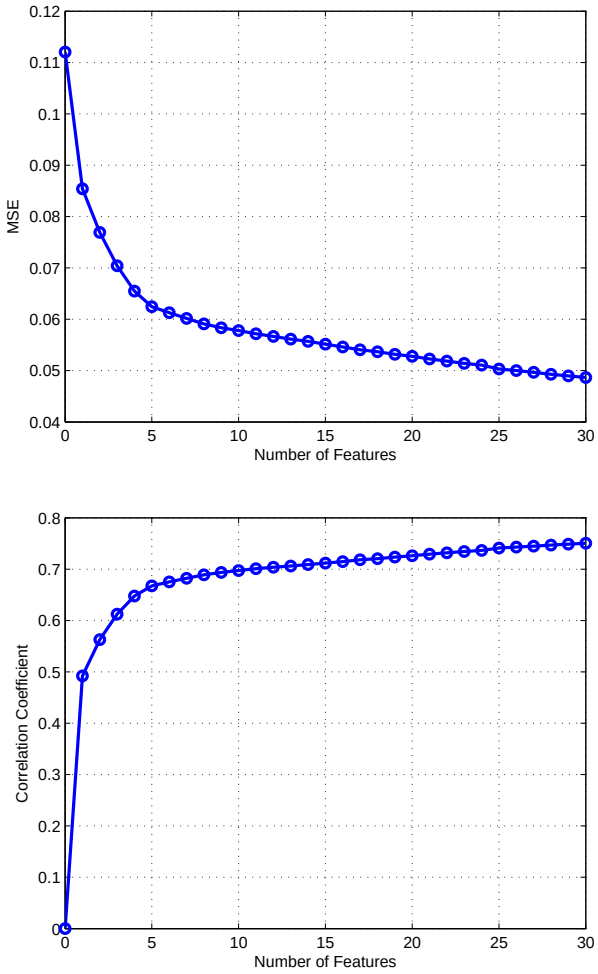


Figure 5.7: Performance of the prediction model

The spatial location of the packet is also critical to the visibility. Viewers are usually attracted by the objects at the center of the screen, both because the camera location is often chosen to place interesting objects at the center, and also because the large screen sizes of HDTV mean the viewer is often less aware of the periphery. A large value of IsColor $\times$ DevFromBorder means the loss affects both views and appears near the center.

The objective quality metrics are also helpful. TMDR $\times$ MaxMSE is the second most important feature. It implies that a big distortion which lasts for a long time is very

Table 5.5: The Ten Most Important Features of the Prediction Model

Feature #	Feature	Coefficient
γ	1	-3.2515
1	IsColor $\times$ CIntraRatio	0.0328
2	TMDR $\times$ MaxMSE	2.3362e-6
3	IsColor $\times$ DevFromBorder	0.0634
4	IsCBframe $\times$ CMaxMotA	-0.5752
5	IsColor $\times$ CMaxMotM	0.0047
6	IsCPframe $\times$ CNumIntra	0.0031
7	D(Skip/Direct)Ratio $\times$ MinSSIM	-1.7107
8	PktSize	0.0012
9	DInterRatio $\times$ DVarMotA	-7.1154
10	IsDBframe $\times$ DMaxMotX	-0.0113

likely to be seen. The feature with MinSSIM carries a negative coefficient, as smaller value of SSIM indicates worse quality.

The frame type is another important factor in the model. The features related to IsCBframe and IsDBframe have negative coefficients as would be expected, since losses in B frames are less visible than average losses. IsCPframe $\times$ CNumIntra has a positive impact on the visibility. A large value of intra-coded macroblocks implies the motion is complicated or there is a scene cut. Then zero motion copy would probably not yield a good result.

The single term PktSize carries a positive sign. Most of the values of PktSize are around 1316. The packet size can be well below 1316 bytes if the whole frame is included in the packet or if the packet is the last one in that frame. In the first scenario, it may imply the residual energy is small and motion is not very complicated. In the second scenario, it means the loss is far away from the center, thus less visible.

5.4 Summary

We present a human observer experiment on fixed-sized packet loss visibility of 2D+depth compressed 3D video. We found that losses in color frames are generally more likely to be seen than losses in depth frames, probably due to the different types of artifacts they cause and the number of views affected by the loss. Losses in color P frames are the most damaging, even worse than losses in color I frames. Among losses in depth frames, I frames are the most difficult to conceal thus are the most visible. We build an encoder-based model to predict the visibility of packet losses with features related to frame type, spatial location of the loss and motion complexity. The model shows good performance in terms of MSE and correlation coefficient.

Acknowledgment

This research was supported in part by the Intel/Cisco Video Aware Wireless Network (VAWN) program, and by the National Science Foundation under grant CCF-1160832.

Chapter 5, in part, is a reprint of material as it appears in Q. Song and P. C. Cosman, "Packet loss visibility of view+depth compressed stereo 3D video", *International Packet Video Workshop*, pp. 1-7, Dec. 2013. The dissertation author was the primary author of this paper and the co-author Prof. Cosman directed and supervised the research which forms the basis of Chapter 5.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

In this dissertation, we studied the enhancement and preservation of perceptual quality of 2D LDR videos adapted to viewing conditions, HDR videos generated by inverse tone mapping, and 2D+depth stereoscopic 3D videos affected by packet losses.

In Chapter 2, we proposed two tone mapping operators to enhance the luminance and details of videos shown in bright ambient illumination. The tone mapping considers display characteristics and human visual sensitivity. The contrast loss in dark areas of videos due to reflected light and reduced sensitivity of eyes is mitigated. The content independent tone mapping operator is constructed only once for the given viewing condition and can be applied to any video. The content dependent method uses simple statistics of a video, and slightly outperforms the content independent method. Our proposed methods boost the visibility of details in dark areas and preserve details well in bright areas.

In Chapter 3, we presented a subjective test which confirms the ability to reduce encoded bit-rate without impacting the perceptual quality by adapting the representation

and encoded bit-rate to the variable viewing conditions. A substantial bit rate savings can be realized if the tablet device can determine the viewing distance and the content delivered to the device is adapted to the distance.

In Chapter 4, we proposed a debanding filter to enhance the perceptual quality of inverse tone mapped HDR videos. Banding artifacts resulting from inverse tone mapping and blocky artifacts resulting from compression are removed, or at least greatly reduced. The filter combines non-smooth area detection and filtering, and is able to preserve edges and details. The parameters of the filter are selected by finding a trade-off between banding removal and detail preservation. The filter works much more efficiently than the debanding algorithms and edge-preserving filters in the literature. Subjective tests demonstrate the performance of our proposed filter.

In Chapter 5, we presented a human observer experiment on fixed-sized packet loss visibility of 2D+depth compressed 3D video. We found that color frames are generally more important than depth frames, because color frames affect both views, while depth frames only affect one view. The importance of packets also depends on the frame type, spatial location of the packet and motion complexity, etc. A prediction model of the packet importance (loss visibility) is built using features extracted from the video. The model can be used for unequal error protection in the tranmission of 2D+depth stereoscopic 3D video.

6.2 Future Work

In Chapter 2, the contrast of the codewords with higher histogram counts is enhanced more by the proposed content dependent enhancement method, since these codewords take larger areas in the video. In addition to this “spatial weighting”, a “temporal weighting” can be incorporated by considering the motion of objects, as

moving objects are more likely to draw the viewer's attention. Motion can be estimated by the motion vectors collected in the video decoding process. Temporal weighting factors can be computed as a function of the average motion vector of each codeword.

Note that the enhancement methods in Chapter 2 assume that the device has the ability to generate and apply the tone mapping operators. In the situation where the device does not have such ability and it can only show the received videos, many details can be invisible. In [27, 75, 76, 77], bit-rate saving by ambient light adaptation was investigated. However, none of these works studied the greater effects of ambient light on the dark areas of videos than on the bright areas. In future work, bit-rate saving by considering the more severe degradation on the dark areas can be explored. It can be achieved by filtering before compression, or allocating fewer bits to dark areas of videos during compression.

Moreover, a scalable coding structure by ambient light adaptation can be built. Each layer provides a video version corresponding to an ambient illumination. Say there are four layers for 10,000 lx, 5000 lx, 500 lx and 0 lx. The base layer is the video version corresponding to 10,000 lx. It includes very coarse details, but has the same perceptual quality under 10,000 lx as the original video displayed in the dark. The enhancement layer 1 includes the difference of the video version of 5000 lx from the base layer, and the enhancement layer 2 includes the difference of the video version of 500 lx from the video reconstructed from the enhancement layer 1, etc. Hence, only the base layer and the necessary enhancement layers need to be transmitted to the viewer, according to the ambient light level.

HDR videos are expected to maintain or increase in popularity for several years. The first HDR cinema was open in 2015. Consumer HDR televisions have emerged recently. Mobile devices would support HDR videos in the near future. Viewing conditions also play a role in the perceptual quality of HDR. Since HDR displays have darker blacks

than LDR displays, the ambient illumination can have greater impact on HDR videos. The methods proposed in Chapter 2 can be extended to HDR.

Another popular topic is virtual reality (VR), which gives a 360-degree 3D environment. The viewing range is much wider than that of stereoscopic 3D. Depth maps are likely to be used in the compression of VR to achieve low bit-rate. The importance of depth maps on VR can be higher than that on stereoscopic 3D videos, as the depth perception is emphasized in VR. In other words, packet losses in depth maps can have higher impact on the perceptual quality. This would be interesting to explore in the future.

Appendix A

Proof of Relationship between Span and Output of Sparse Filter

For uniform banding steps, a 5-tap unweighted sparse filter (no decision process) with equidistant samples can create at most four new codewords at each banding step in one direction. The proof is as follows.

The input uniform banding signal is represented as:

$$x[m,n] = p + \lfloor \frac{n-n_0}{W} \rfloor \cdot H, \tag{A.1}$$

where $W \geq 2$ is the width of each banding step, and $H > 0$ is the difference of codewords

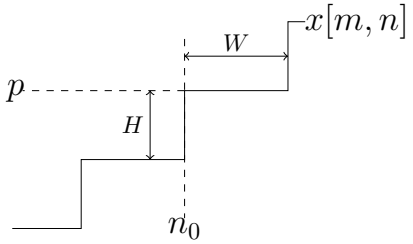


Figure A.1: Input uniform signal.

between adjacent steps (Fig. A.1). Now we derive the output of horizontal filtering at $n_0 \leq n \leq n_0 + W - 1$ where $n_0 \geq 2$. The four samples entering the filter in addition to the sample at $x[m,n]$ are: $x[m,n-2D] = p - aH$, $x[m,n-D] = p - bH$, $x[m,n+D] = p + cH$, and $x[m,n+2D] = p + dH$, where $a,b,c,d \in \mathbb{N}^0$. We will prove two properties: 1) $a-1 \leq d \leq a+1$, and 2) $\lfloor \frac{a}{2} \rfloor \leq b \leq \lceil \frac{a}{2} \rceil$.

1) $a-1 \leq d \leq a+1$: from (A.1), we obtain:

$$x[m,n-2D] = p + \lfloor \frac{n-2D-n_0}{W} \rfloor \cdot H. \tag{A.2}$$

Since $x[m,n-2D] = p - aH$, we obtain:

$$p - aH = p + \lfloor \frac{n-2D-n_0}{W} \rfloor \cdot H, \tag{A.3}$$

$$\Rightarrow -aW \leq n - 2D - n_0 \leq -aW + W - 1, \tag{A.4}$$

$$\Rightarrow n - n_0 + aW - W + 1 \leq 2D \leq n - n_0 + aW. \tag{A.5}$$

Add n to the inequality (A.5):

$$2n - n_0 + (a-1)W + 1 \leq n + 2D \leq 2n - n_0 + aW. \tag{A.6}$$

Since n is in the range: $n_0 \leq n \leq n_0 + W - 1$, we obtain:

$$\begin{cases} n + 2D \geq 2n - n_0 + (a-1)W + 1 \geq n_0 + (a-1)W + 1 \\ n + 2D \leq 2n - n_0 + aW \leq n_0 + 2W - 2 + aW \end{cases} \tag{A.7}$$

Combining both these inequalities with (A.1) and the fact that $x[m,n]$ is non-decreasing,

we obtain:

$$p + \lfloor a - 1 + \frac{1}{W} \rfloor \cdot H \leq x[m, n+2D] \leq p + \lfloor 2 - \frac{2}{W} + a \rfloor \cdot H. \tag{A.8}$$

Since $x[m, n+2D] = p + dH$,

$$p + \lfloor a - 1 + \frac{1}{W} \rfloor \cdot H \leq p + dH \leq p + \lfloor 2 - \frac{2}{W} + a \rfloor \cdot H. \tag{A.9}$$

As $W \geq 2$, we obtain

$$a - 1 \leq d \leq a + 1. \tag{A.10}$$

2) $\lfloor \frac{a}{2} \rfloor \leq b \leq \lceil \frac{a}{2} \rceil$: from (A.5), we obtain:

$$\frac{-n + n_0 - aW}{2} \leq -D \leq \frac{-n + n_0 - aW + W - 1}{2}. \tag{A.11}$$

Add n to the inequality (A.11):

$$\frac{n + n_0 - aW}{2} \leq n - D \leq \frac{n + n_0 - aW + W - 1}{2}. \tag{A.12}$$

Since $n_0 \leq n \leq n_0 + W - 1$:

$$n_0 - \frac{aW}{2} \leq n - D \leq n_0 - \frac{aW}{2} + W - 1. \tag{A.13}$$

Combining (A.13) with (A.1) and the fact that $x[m, n]$ is non-decreasing, we obtain:

$$p + \lfloor \frac{-a}{2} \rfloor \cdot H \leq x[m, n-D] \leq p + \lfloor -\frac{a}{2} + 1 - \frac{1}{W} \rfloor \cdot H. \tag{A.14}$$

Table A.1: All the possible combinations of b, c and d when $a = 2K+1$ where $K \in \mathbb{N}^0$.

b	c	d	Output	Conditions of D
K	K	$2K$	$p-\frac{1}{5}H$	$0 < D-KW < \frac{W}{3}$
$K+1$	K	$2K$	$p-\frac{2}{5}H$	$0 < D-KW < \frac{W}{2}$
K	K	$2K+1$	p	$\frac{W}{4} < D-KW < \frac{W}{2}$
K	$K+1$	$2K+1$	$p+\frac{1}{5}H$	$\frac{W}{3} < D-KW < \frac{2W}{3}$
$K+1$	K	$2K+1$	$p-\frac{1}{5}H$	$\frac{W}{3} < D--KW < \frac{2W}{3}$
$K+1$	$K+1$	$2K+1$	p	$\frac{W}{2} < D-KW < \frac{3W}{4}$
K	$K+1$	$2K+2$	$p+\frac{2}{5}H$	$\frac{W}{2} < D-KW < W$
$K+1$	$K+1$	$2K+2$	$p+\frac{1}{5}H$	$\frac{2W}{3} < D-KW < W$

Since $x[m,n-D] = p-bH$,

$$\begin{aligned} & p+\lfloor\frac{-a}{2}\rfloor\cdot H \le p-bH \le p+\lfloor-\frac{a}{2}+1-\frac{1}{W}\rfloor\cdot H \\ \Rightarrow & -\lfloor-\frac{a}{2}+1-\frac{1}{W}\rfloor \le b \le -\lfloor-\frac{a}{2}\rfloor. \end{aligned} \tag{A.15}$$

Since $W \ge 2$, we obtain:

$$\lfloor\frac{a}{2}\rfloor \le b \le \lceil\frac{a}{2}\rceil. \tag{A.16}$$

Similarly, we can prove that

$$\lfloor\frac{d}{2}\rfloor \le c \le \lceil\frac{d}{2}\rceil. \tag{A.17}$$

When a is odd, there are 8 possible combinations of b, c and d that satisfy (A.10), (A.16) and (A.17). The combinations are shown in Table A.1 where a is represented as $2K+1$ with $K \in \mathbb{N}^0$. When a is even, there are only 5 possible combinations of b, c and d that satisfy the properties. The combinations are shown in Table A.2 where $a = 2K$ for $K \in \mathbb{N}^0$. "Output" in the tables means the filtering output, $\frac{1}{5}\sum_{-2}^{2} x[m,n+jD]$. The output has 5 possible values: p, $p-\frac{2}{5}H$, $p-\frac{1}{5}H$, $p+\frac{1}{5}H$ and $p+\frac{2}{5}H$. Therefore, a 5-tap sparse filter with fixed equidistant sample spacing can generate at most four new codewords when used on equi-width banding steps.

The range of D has to satisfy the conditions listed in Tables A.1 and A.2 so that

Table A.2: All the possible combinations of b, c and d when $a = 2K$ where $K \in \mathbb{N}^0$.

b	c	d	Output	Conditions of D
K	$K-1$	$2K-1$	$p-\frac{2}{5}H$	$-\frac{W}{2} < D-KW < 0\ (K>0)$
K	K	$2K-1$	$p-\frac{1}{5}H$	$-\frac{W}{3} < D-KW < 0\ (K>0)$
K	K	$2K$	p	$-\frac{W}{4} < D-KW < \frac{W}{4}$
K	K	$2K+1$	$p+\frac{1}{5}H$	$0 < D-KW < \frac{W}{3}$
K	$K+1$	$2K+1$	$p+\frac{2}{5}H$	$0 < D-KW < \frac{W}{2}$

there are $n \in [n_0, n_0+W-1]$ that can achieve the combination. The ranges of D overlap, so some values of D can generate as many as four new codewords. Table A.3 shows the output codewords and the corresponding widths of the mini-steps for different ranges of D. For simplicity, D' is used to represent $D-KW$. The widths of mini-steps are computed by determining the range of n for each combination of the input codewords. Zero width means the codeword cannot be generated by this range of D. In most of the circumstances, four new codewords can be generated by the filter. Fewer than 4 new codewords will be created only when $D = KW + \frac{K'}{q}W$ where $q = 4$ or 3 and $K' \in \mathbb{Z}$. Note that D is an integer, so these values of D can be achieved only when W or K' is a multiple of q. The table indicates that D and $D+KW$ yield exactly the same filtering output. It also indicates that the width of the output mini-steps is greater than or equal to $\frac{W}{5}$, where the minima occur at $D = \frac{K'W}{5} + KW$, where $K' \in \mathbb{Z}$ and K' is not a multiple of 5.

Table A.3: Widths of output mini-steps after filtering for different ranges of $D' = D - KW$ where $K \in \mathbb{N}^0$.

Range of D'	Widths of mini-steps at each codeword				
	$p-\frac{2}{5}H$	$p-\frac{1}{5}H$	p	$p+\frac{1}{5}H$	$p+\frac{2}{5}H$
$0 < D' < \frac{W}{4}$	D'	D'	$W-4D'$	D'	D'
$D' = \frac{W}{4}$	D'	D'	0	D'	D'
$\frac{W}{4} < D' < \frac{W}{3}$	D'	$W-3D'$	$4D'-W$	$W-3D'$	D'
$D' = \frac{W}{3}$	D'	0	$4D'-W$	0	D'
$\frac{W}{3} < D' < \frac{W}{2}$	$W-2D'$	$3D'-W$	$W-2D'$	$3D'-W$	$W-2D'$
$D' = \frac{W}{2}$	0	$3D'-W$	0	$3D'-W$	0
$\frac{W}{2} < D' < \frac{2W}{3}$	$2D'-W$	$2W-3D'$	$2D'-W$	$2W-3D'$	$2D'-W$
$D' = \frac{2W}{3}$	$2D'-W$	0	$2D'-W$	0	$2D'-W$
$\frac{2W}{3} < D' < \frac{3W}{4}$	$W-D'$	$3D'-2W$	$3W-4D'$	$3D'-2W$	$W-D'$
$D' = \frac{3W}{4}$	$W-D'$	$3D'-2W$	0	$3D'-2W$	$W-D'$
$\frac{3W}{4} < D' < W$	$W-D'$	$W-D'$	$4D'-3W$	$W-D'$	$W-D'$
$D' = W$	0	0	W	0	0

Bibliography

[1] G. J. Sullivan, J. R. Ohm, W. J. Han, and T. Wiegand, "Overview of the high efficiency video coding (HEVC) standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1649–1668, Dec 2012.

[2] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 560–576, July 2003.

[3] Y.-L. Chang, T.-L. Lin, and P. Cosman, "Network-based IP packet loss importance model for H.264 SD videos," in *IEEE 18th International Packet Video Workshop (PV)*, 2010, pp. 178–185.

[4] T. L. Lin, S. Kanumuri, Y. Zhi, D. Poole, P. C. Cosman, and A. R. Reibman, "A versatile model for packet loss visibility and its application to packet prioritization," *IEEE Transactions on Image Processing*, vol. 19, no. 3, pp. 722–735, March 2010.

[5] Y. L. Chang, T. L. Lin, and P. C. Cosman, "Network-based H.264/AVC whole-frame loss visibility model and frame dropping methods," *IEEE Transactions on Image Processing*, vol. 21, no. 8, pp. 3353–3363, Aug 2012.

[6] R. Vanam and Y. Reznik, "Perceptual pre-processing filter for user-adaptive coding and delivery of visual information," in *Picture Coding Symposium (PCS), 2013*, Dec 2013, pp. 426–429.

[7] S. Miller, M. Nezamabadi, and S. Daly, "Perceptual signal coding for more efficient usage of bit codes," in *Annual Technical Conference Exhibition, SMPTE 2012*, Oct 2012, pp. 1–9.

[8] F. Dufaux, P. L. Callet, R. Mantiuk, and M. Mrak, *High Dynamic Range Video: From Acquisition, to Display and Applications*. Academic Press, 2016.

[9] 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, April 2004.

[10] "ITU-R Recommendation BT.500-13: Methodology for the subjective assessment of the quality of television pictures," January 2012.

[11] "ITU-T Recommendation P.910: Subjective video quality assessment methods for multimedia applications," April 2008.

[12] J. H. Krantz, L. D. Silverstein, and Y.-Y. Yeh, "Visibility of transmissive liquid crystal displays under dynamic lighting conditions," *Human Factors*, vol. 34, no. 5, pp. 615–632, Oct. 1992.

[13] M. Miller and J. Niederbaumer, "Automatic luminance and contrast adjustment as functions of ambient/surround luminance for display device," U.S. Patent 6,529,212 B2, 2003.

[14] Display Mate website: http://www.displaymate.com.

[15] R. Mantiuk, S. Daly, and L. Kerofsky, "Display adaptive tone mapping," in *ACM SIGGRAPH 2008 papers (SIGGRAPH '08)*, 2008.

[16] H. Kobiki and M. Baba, "Preserving perceived brightness of displayed image over different illumination conditions," in *2010 IEEE International Conference on Image Processing*, Sept 2010, pp. 2485–2488.

[17] M.-Y. Lee, C.-H. Son, J.-M. Kim, C.-H. Lee, and Y.-H. Ha, "Illumination-level adaptive color reproduction method with lightness adaptation and flare compensation for mobile display," *Journal of Imaging Science and Technology*, no. 1, pp. 44–52, Jan - Feb 2007.

[18] Y. J. Kim, "An automatic image enhancement method adaptive to the surround luminance variation for small sized mobile transmissive LCD," *IEEE Transactions on Consumer Electronics*, vol. 56, no. 3, pp. 1161–1166, Aug 2010.

[19] X. Xu and L. Kerofsky, "Improving content visibility for high-ambient-illumination viewable display and energy-saving display," *Journal of the Society for Information Display*, vol. 19, no. 9, pp. 645–654, 2011.

[20] H. Su, C. Jung, S. Wang, and Y. Du, "Adaptive enhancement of luminance and details in images under ambient light," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, March 2016, pp. 1219–1223.

[21] "ITU-R Recommendation BT.1886: Reference electro-optical transfer function for flat panel displays used in HDTV studio production," March 2011.

[22] "ITU-R Report BT.2246: The present state of ultra high definition television," October 2011.

[23] P. G. J. Barten, "Formula for the contrast sensitivity of the human eye," in *SPIE-IS&T*, January 2004.

[24] The video sequences are from: http://5.co.il/kimono1-and-park-scene-test-results/.

[25] The video Sequences are from: https://media.xiph.org/video/derf/.

[26] J. Young, M. Trudeau, D. Odell, K. Marinelli, and J. Dennerlein, "Touch-screen tablet user configurations and case-supported tilt affect head and neck flexion angles," *Work: A Journal of Prevention, Assessment and Rehabilitation*, vol. 41, no. 1, pp. 81–91, 2012.

[27] J. Xue and C. W. Chen, "Mobile video perception: New insights and adaptation strategies," *IEEE Journal of Selected Topics in Signal Processing*, vol. 8, no. 3, pp. 390–401, June 2014.

[28] Y. Reznik, E. Asbun, Z. Chen, Y. Ye, E. Zeira, R. Vanam, Z. Yuan, G. Sternberg, A. Zeira, and N. Soni, "User-adaptive mobile video streaming," in *Visual Communications and Image Processing, 2012 IEEE*, Nov 2012.

[29] Y. Reznik, "User-adaptive mobile video streaming using MPEG-DASH," in *SPIE Optical Engineering+ Applications*. International Society for Optics and Photonics, 2013, pp. 88 560J–88 560J.

[30] "x264," http://www.videolan.org/developers/x264.html.

[31] J.-S. Lee, F. De Simone, and T. Ebrahimi, "Video coding based on audio-visual attention," in *Multimedia and Expo, IEEE Intl. Conference on*, June 2009, pp. 57–60.

[32] "ITU-R Recommendation BT.709-6: Parameter values for the HDTV standards for production and international programme exchange," June 2015.

[33] E. Reinhard, W. Heidrich, P. Debevec, S. Pattanaik, G. Ward, and K. Myszkowski, *High Dynamic Range Imaging: Acquisition, Display, and Image-Based Lighting*, 2nd ed. Morgan Kaufmann, 2010.

[34] "SMPTE recommended practice: D-Cinema quality - reference projector and environment," *SMPTE RP 431-2:2011*, pp. 1–14, April 2011.

[35] 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," in *ACM SIGGRAPH*, 2007.

[36] F. Banterle, P. Ledda, K. Debattista, and A. Chalmers, "Inverse tone mapping," in *Proc. the 4th International Conference on Computer Graphics and Interactive Techniques in Australasia and Southeast Asia*, New York, NY, USA, 2006, pp. 349–356.

[37] P.-H. Kuo, C.-S. Tang, and S.-Y. Chien, "Content-adaptive inverse tone mapping," in *Proc. IEEE International Conference on Visual Communications and Image Processing (VCIP)*, Nov 2012, pp. 1–6.

[38] Q. Chen, G.-M. Su, and P. Yin, "Near constant-time optimal piecewise LDR to HDR inverse tone mapping," in *Proc. SPIE*, vol. 9404, 2015, pp. 94 040O–11.

[39] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 3rd ed. Pearson, 2007.

[40] H. Foley and M. Matlin, *Sensation and Perception*, 5th ed. Psychology Press, 2009.

[41] "ITU-R Recommendation BT.601-7: Studio encoding parameters of digital television for standard 4:3 and wide-screen 16:9 aspect ratios," March 2011.

[42] W. Ahn and J.-S. Kim, "Flat-region detection and false contour removal in the digital TV display," in *Proc. IEEE International Conference on Multimedia and Expo*, July 2005, pp. 1338–1341.

[43] K. Yoo, H. Song, and K. Sohn, "In-loop selective processing for contour artefact reduction in video coding," *Electronics Letters*, vol. 45, no. 20, pp. 1020–1022, September 2009.

[44] G.-M. Su, T. Chen, P. Yin, and S. Qu, "Guided post-prediction filtering in layered VDR coding," U.S. Patent 8,897,581 B2, Nov. 25, 2014.

[45] G.-M. Su, S. Qu, and S. Daly, "Adaptive false contouring prevention in layered coding of images with extended dynamic range," U.S. Patent 8,873,877 B2, Oct. 28, 2014.

[46] S. J. Daly and X. Feng, "Decontouring: prevention and removal of false contour artifacts," in *Proc. SPIE*, vol. 5292, 2004, pp. 130–149.

[47] J. W. Lee, B. R. Lim, R.-H. Park, J.-S. Kim, and W. Ahn, "Two-stage false contour detection using directional contrast and its application to adaptive false contour reduction," *IEEE Transactions on Consumer Electronics*, vol. 52, no. 1, pp. 179–188, Feb 2006.

[48] S. Bhagavathy, J. Llach, and J. Zhai, "Multi-scale probabilistic dithering for suppressing banding artifacts in digital images," in *Proc. IEEE International Conference on Image Processing*, vol. 4, Sept 2007, pp. IV–397 – IV–400.

[49] Q. Huang, H. Y. Kim, W. J. Tsai, S. Y. Jeong, J. S. Choi, and C. C. J. Kuo, "Understanding and removal of false contour in HEVC compressed images," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. PP, no. 99, pp. 1–1, 2016.

[50] Y. Niu, X. Wu, and G. Shi, "Image enhancement by entropy maximization and quantization resolution upconversion," *IEEE Transactions on Image Processing*, vol. 25, no. 10, pp. 4815–4828, Oct 2016.

[51] Y. Neuvo, C.-Y. Dong, and S. Mitra, "Interpolated finite impulse response filters," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 32, no. 3, pp. 563–570, Jun 1984.

[52] Y. C. Lim, H. K. Kwan, and W.-C. Siu, *Trends in Digital Signal Processing: A Festschrift in Honour of A.G. Constantinides*. Pan Stanford, 2015.

[53] P. Charbonnier, L. Blanc-Feraud, G. Aubert, and M. Barlaud, "Deterministic edge-preserving regularization in computed imaging," *IEEE Transactions on Image Processing*, vol. 6, no. 2, pp. 298–311, Feb 1997.

[54] N. Azzabou, N. Paragios, F. Guichard, and F. Cao, "Variable bandwidth image denoising using image-based noise models," in *2007 IEEE Conference on Computer Vision and Pattern Recognition*, June 2007, pp. 1–7.

[55] Z. Farbman, R. Fattal, D. Lischinski, and R. Szeliski, "Edge-preserving decompositions for multi-scale tone and detail manipulation," in *ACM SIGGRAPH 2008 Papers*, 2008, pp. 67:1–67:10.

[56] D. Lischinski, Z. Farbman, M. Uyttendaele, and R. Szeliski, "Interactive local adjustment of tonal values," in *ACM SIGGRAPH 2006 Papers*, 2006, pp. 646–653.

[57] M. Nikolova, "Minimizers of cost-functions involving nonsmooth data-fidelity terms. application to the processing of outliers," *SIAM Journal on Numerical Analysis*, vol. 40, no. 3, pp. 965–994, 2002.

[58] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *Sixth International Conference on Computer Vision*, Jan 1998, pp. 839–846.

[59] D. Scharstein and R. Szeliski. Middlebury stereo evaluation. [Online]. Available: http://vision.middlebury.edu/stereo/

[60] Y. Boykov and V. Kolmogorov, "An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 9, pp. 1124–1137, 2004.

[61] M. Tanimoto, T. Fujii, K. Suzuki, N. Fukushima, and Y. Mori, *Reference Softwares for Depth Estimation and View Synthesis*, ISO/IEC JTC1/SC29/WG11 MPEG 2008/M15377, April 2008.

[62] P. Merkle, A. Smolic, K. Müller, and T. Wiegand, "Multi-view video plus depth representation and coding," in *IEEE International Conference on Image Processing*, vol. 1, 2007, pp. I – 201–I – 204.

[63] C. Fehn, "Depth-image-based rendering (DIBR), compression, and transmission for a new approach on 3D-TV," in *SPIE Stereoscopic Displays and Virtual Reality Systems XI*, vol. 5291, 2004, pp. 93–104.

[64] E. Bosc, R. Pépion, P. Le Callet, M. Köppel, P. Ndjiki-Nya, L. Morin, and M. Pressigout, "Perceived quality of dibr-based synthesized views," in *SPIE Applications of Digital Image Processing XXXIV*, vol. 8135, 2011.

[65] A. Jain, L. Tran, R. Khoshabeh, and T. Nguyen, "Efficient stereo-to-multiview synthesis," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2011, pp. 889–892.

[66] A. Telea, "An image inpainting technique based on the fast marching method," *Journal of Graphics Tools*, vol. 9, no. 1, pp. 23–34, 2004.

[67] C. T. E. R. Hewage, S. Worrall, S. Dogan, H. Kodikaraarachchi, and A. Kondoz, "Stereoscopic TV over IP," in *4th European Conference on Visual Media Production*, 2007, pp. 1–7.

[68] C. T. E. R. Hewage, S. Worrall, S. Dogan, S. Villette, and A. Kondoz, "Quality evaluation of color plus depth map-based stereoscopic video," *IEEE Journal of Selected Topics in Signal Processing*, vol. 3, no. 2, pp. 304–318, 2009.

[69] L. Pinto, J. Carreira, S. Faria, N. Rodrigues, and P. Assuncao, "Subjective quality factors in packet 3D video," in *2011 Third International Workshop on Quality of Multimedia Experience (QoMEX)*, 2011, pp. 149–154.

[70] "H.264/AVC JM reference software," http://iphome.hhi.de/suehring/tml/.

[71] *Common Test Conditions for 3DV experimentation*, ISO/IEC JTC1/SC29/WG11 MPEG 2011/N12745, May 2012.

[72] *DSL Forum Technical Report TR-126: Triple-play Services Quality of Experience (QoE) Requirements*, December 2006.

[73] W.-M. Lam, A. Reibman, and B. Liu, "Recovery of lost or erroneously received motion vectors," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 5, 1993, pp. 417–420.

[74] Y. Liu, J. Wang, and H. Zhang, "Depth image-based temporal error concealment for 3-D video transmission," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 20, no. 4, pp. 600–604, 2010.

[75] G. Nur, H. K. Arachchi, S. Dogan, and A. M. Kondoz, "Ambient illumination as a context for video bit rate adaptation decision taking," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 20, no. 12, pp. 1887–1891, Dec 2010.

[76] ——, “Extended VQM model for predicting 3D video quality considering ambient illumination context,” in *2011 3DTV Conference: The True Vision - Capture, Transmission and Display of 3D Video (3DTV-CON)*, May 2011, pp. 1–4.

[77] F. C. Nur and G. Nur, “Prediction of 3D video experience from video quality and depth perception considering ambient illumination context,” in *The First International Conference on Future Generation Communication Technologies*, Dec 2012, pp. 28–31.