



MONASH University

**A Study on Augmenting HDR Image by Extra Features
through Data Hiding**

Chia Kai Lin

Doctor of Philosophy

A thesis submitted for the degree of Doctor of Philosophy at

Monash University in 2021

School of Information Technology

Copyright notice

©Chia Kai Lin (2021)

I certify that I have made all reasonable efforts to secure copyright permissions for third-party content included in this thesis and have not knowingly added copyright content to my work without the owner's permission.

Abstract

The recent advancement in hardware and software technologies enables users to conveniently create, modify and store various digital contents, which can then be shared with peers. Among them, digital image appears to be the most popular type of content due to its versatility. In fact, since the inception of digital imaging, it has been utilised in different applications to further improve the quality of life. Consumers can also use digital camera to record precious moments in their daily life. Recently, a technique called high dynamic range (HDR) imaging is getting popular. HDR imaging can capture and display significantly more colours when compared to the legacy low dynamic range technology, which can only handle 8-bit per channel. Thanks to the advancement of smart devices, nowadays HDR images can be captured even with an entry-level smartphone. The increasing popularity of HDR imaging also leads to the increasing trend of storing HDR images of different formats (such as OpenEXR) on devices or on the cloud, where ownership and integrity of the contents are potentially compromised. Data hiding, which inserts data into a content to serve certain purpose, is known to be an effective solution for integrity checking through fragile watermarking, hyperlinking related contents, as well as providing specialized or extra features. Furthermore, image masking, which can mask the semantic of an image, has also been heavily relied on to preserve privacy. Therefore, in this thesis, two reversible data hiding methods are developed for OpenEXR image. Their performances are evaluated then compared with existing methods. Subsequently, as a means to improve the management of the OpenEXR image, a reversible data hiding method in masked OpenEXR image is proposed. Finally, for backward compatibility and adaptation to future display technologies, scalable coding of HDR image is

investigated to provide progression quality improvement in terms of dynamic range by using the newly rearranged file structure. Each proposed method is discussed in detail in the Chapters of this thesis.

Declaration

This thesis contains no material which has been accepted for the award of any other degree or diploma at any university or equivalent institution and that, to the best of my knowledge and belief, this thesis contains no material previously published or written by another person, except where due reference is made in the text of the thesis.

Signature:

Name:

Date:

Publications during enrolment

Chapter 3:

- KaiLin Chia, KokSheik Wong and Jean-Luc Dugelay, "High Capacity Reversible Data Embedding for OpenEXR Image", **in preparation** for submission to Multimedia Tools and Applications.

Chapter 4:

- KaiLin Chia, KokSheik Wong and Jean-Luc Dugelay, "Data Hiding in Perceptually Masked OpenEXR Image," 2019 IEEE 21st International Workshop on Multimedia Signal Processing (MMSP), 2019, pp. 1-6.

Acknowledgement

Firstly, I am eternally grateful to my supervisor A/Prof Dr Wong Kok Sheik, for his inspired guidance and advice, his infinite patience, encouragement and support during my PhD years. I am greatly indebted to him for his kindness and understanding, in both academic and personal aspects.

I am also very thankful to Prof Raphael Phan, for his input on security analysis.

I would also like to express my gratitude to Prof Jean-Lun Dugelay from EURECOM, France for his advice and guidance during my visit to the institute.

Next, my appreciative thoughts go to my father, my mother and my siblings for their great love, support and encouragement during my PhD journey. Thank you so much for always being there for me.

I would like to thank Monash University Malaysia for the administrative and financial support throughout my candidature.

I would also like to thank The Ministry of Energy, Science, Technology, Environment and Climate Change (MESTECC) for funding this research under the E-Science grant (01-02-10-SF0327).

Lastly, I would like to thank EU Horizon 2020 - Marie Skłodowska-Curie Actions (MSCA) for funding my visit to EURECOM and other research related activities through the project entitled Computer Vision Enabled Multimedia Forensics and People Identification (Project No. 690907, Acronym: IDENTITY).

Contents

1	Introduction	1
1.1	Problem statement	5
1.2	Research questions	6
1.3	Research objectives	6
1.4	Scope of project	7
1.5	Structure of the thesis	8
2	Literature review	10
2.1	Human visual system	10
2.2	High dynamic range imaging	12
2.2.1	HDR image acquisition	14
2.2.2	Representation of HDR image pixels	15
2.2.3	HDR file formats	17
2.2.4	File format chosen	22
2.3	Display devices	23
2.3.1	Cathode Ray Tube display	24
2.3.2	Liquid Crystal Display	25
2.3.3	Plasma display	27
2.3.4	Light Emitting Diode display	27

2.3.5	Organic Light Emitting Diode (OLED) display	28
2.3.6	HDR display devices	29
2.4	HDR image quality assessment	31
2.4.1	Luminance-independent metrics	34
2.4.2	Display-referred metrics	35
2.5	Tone mapping operator	36
2.5.1	Inverse tone mapping operator	38
2.6	Data hiding	40
2.6.1	Traditional data hiding	44
2.6.2	HDR data hiding	51
2.7	Image masking	60
2.7.1	Joint data hiding and image masking	61
2.8	Scalable coding	62
2.9	Summary	63
3	Reversible Data Hiding in OpenEXR image	67
3.1	Histogram Shifting based Reversible Data Hiding	68
3.1.1	Experiments and discussions	73
3.1.2	Summary and conclusion	79
3.2	Reversible Data Hiding based on Natural Characteristics of Pixels	79
3.2.1	Experiments and discussions	83
3.2.2	Summary and conclusion	87
3.3	Application of the proposed method	87
4	Data Hiding in Masked OpenEXR image	91
4.1	Proposed method	92
4.1.1	Preprocessing	92

4.1.2	Masking The Image	94
4.1.3	Data Hiding	95
4.1.4	Data Extraction and Image Recovery	96
4.2	Experiments and Evaluations	97
4.2.1	Effective Payload	98
4.2.2	Image Quality	100
4.2.3	Randomness test	101
4.2.4	Key sensitivity test	103
4.3	Discussion	105
4.4	Summary and conclusion	107
5	Scalable Coding for Backward Compatibility	109
5.1	Proposed method	109
5.2	Results and Discussion	113
5.3	Summary and Conclusions	118
6	Conclusion	119
6.1	Summary	119
6.2	Contributions	120
6.3	Limitations	121
6.4	Future research	121

List of Figures

1.0.1 Five dimensions of image quality improvement [1]	3
1.5.1 Overall structure of the thesis	9
2.1.1 The human eye [2]	11
2.1.2 Dynamic range [3]	12
2.2.1 Visible colour gamut and luminance range [4].	13
2.2.2 HDR imaging pipeline, starting from image acquisition to reproduction [4]	15
2.2.3 An exposure stack of images captured at different exposure levels [5]. .	16
2.2.4 The HDR image resulting from multiple exposure blending. The HDR image is tonemapped for display purposes [5].	16
2.2.5 <i>RGBE encoding</i>	18
2.2.6 <i>32-bit LogLUV bit distribution</i>	19
2.2.7 <i>Illustration of a 16-bit pixel in the OpenEXR half-precision file format. Reproduced from [6]</i>	21
2.2.8 <i>JPEG-Xt decoder</i> [7]	23
2.3.1 CRT display device	25
2.3.2 Working structure of LCD [8]	26
2.3.3 Structure of plasma display [9]	28
2.3.4 Structure of OLED. [10]	29
2.3.5 Structure of first proposed HDR display. [11]	31
2.3.6 The flow of processes for driving the low-resolution backlight modulator and the high-resolution front LCD panel in HDR display [4,11]	32
2.3.7 World first HDR display, DR37-P [12].	32
2.5.1 Examples of tone-mapped images	39
2.5.2 Banding due to unsuitable dynamic range. [13]	40
2.6.1 Data hiding and extraction process	41
2.6.2 Different types of data hiding	41
2.6.3 Example inserting ‘101’ to LSBs.	46
2.6.4 Distortionless data hiding in transform domain [14].	50
2.6.5 IWT-HS method for encrypted images [15].	51
2.6.6 Block diagram of Fujiyoshi <i>et al.</i> ’s RDH method [16].	53
2.6.7 Example of tone mapped original and marked images from [16].	55
2.6.8 Model and visualization of the carriers [17].	58
2.7.1 Example encrypted and decrypted images [18].	60

2.8.1 HDR scalable coding pipeline [19]	64
2.8.2 Decoding paths of the compressed file [19]	64
2.8.3 The baseline method. The HDR image is reconstructed without any loss. Reproduced from [20]	65
2.8.4 The proposed type I method. The bit depth of the residual e_R is reduced by the reversible logarithmic mapping Rev . Reproduced from [20]	65
3.1.1 Example images of different displaying methods.	68
3.1.2 Proposed method flow.	69
3.1.3 Illustration of histogram shifting in mantissa component (right shifting).	72
3.1.4 Six representative HDR test images from [21]. Each image is tone- mapped here for display purposes.	74
3.1.5 Six representative HDR images with data embedded. Each image is tone-mapped here for display purposes.	75
3.1.6 Capacity of each test images	76
3.1.7 Image quality assessment	78
3.2.1 Flowchart of proposed method	80
3.2.2 Number of bits embeddable into each image	84
3.2.3 Image quality assessment	85
3.2.4 Six representative tonemapped HDR images with data embedded to ob- serve the quality.	86
3.3.1 Data hiding application scenario for tone mapping	89
3.3.2 Capacity required to hide DC coefficient of image difference.	90
4.1.1 Overall process of proposed method.	93
4.1.2 An example output of mixing M_1 and M_2 blocks among pixels (Stage II).	95
4.1.3 Masking pixel using XOR operation (Stage III).	96
4.1.4 Hiding data into preprocessed exponent part.	97
4.2.1 Graph of capacity of each test images	99
4.2.2 Graph of image quality assessment (logPSNR)	101
4.2.3 Output images for both scenario.	102
4.2.4 Number of pixels change rate (NPCR)	104
4.2.5 Graph of logPSNR values of unmasked images from key B . The average value of the logPSNR values is 2.6372.	105
4.2.6 Output images when unmasking using the correct key (key A) and the slight modified key (key B)	106
5.1.1 Six representative HDR images with sign bit set to zero. Each image is tone-mapped for display purposes.	111
5.1.2 Arrangement of bits in both the first and second files (one byte each).	113
5.1.3 Different mode of display for <i>Frontier</i>	114
5.1.4 Different mode of display for <i>Flamingo</i>	115
5.2.1 Scalable coding interface	117

List of Tables

1.1	Element of High Image Quality [3]	2
2.1	Semantic of 16-bit floating point representation in OpenEXR.	21
2.2	Summary of data hiding techniques for HDR image	59
3.1	Number of pixels for each exponent values in <i>AhwahneeGreatLounge</i> . .	71
3.2	Example capacity of six test images.	76
3.3	Quality assessment.	77
3.4	Comparison of methods with proposed.	79
3.5	Comparison of methods.	87
3.6	Bits required to store the DC component of the difference image Δ . .	90
4.1	Example hiding capacity for six representative test images.	100
4.2	Quality assessment after processing (dB).	103

List of Symbols

cd/m^2	Candela per square metre
E	Exponent component in OpenEXR
e_P	Prediction error
L_r	Luminance of reference image
L_t	Luminance of test image
L_{max}	Maximum luminance
L_{min}	Minimum luminance
M	Mantissa component in OpenEXR
N	Total number of pixels in an image
S	Sign component in OpenEXR
CIE	International Commission on Illumination
CRT	Cathode ray tube
CSF	Contrast sensitivity function
CT	Computed tomography
D/A	digital-to-analog
DCT	Discrete Cosine Transform
DE	Difference expansion
DFT	Discrete Fourier transform
DVI	Digital Visual Interface
DWT	Discrete Wavelet Transform

EV	Exposure value
HDR	High dynamic range
HDR-VDP	HDR visible difference predictor
HDR-VQM	HDR video quality metric
HDRI	High dynamic range imaging
HS	Histogram shifting
HVS	Human Visual System
ILM	Industrial Light & Magic
ITMO	Inverse tone mapping operator
IWT	Integer wavelet transform
JPEG	Joint Photographic Experts Group
JPEG XT	JPEG extension
LCD	Liquid crystal display
LDR	Low Dynamic Range
LED	Light Emitting Diode
LSB	Least significant bit
MED	Median edge detection
MSE	Mean-square error
OLED	Organic light emitting diode
PEE	Prediction-error expansion
PSNR	Peak signal-to-noise ratio
PU	Perceptual uniform
RDH	Reversible data hiding
SDR	Standard Dynamic Range
TMO	Tone mapping operator
VGA	Video Graphics Array

Chapter 1

Introduction

Digital image is arguably the mostly generated and communicated medium nowadays thanks to the infiltration of smart devices into our everyday lives and the existence of ubiquitous network connection. Digital imaging is the creation of a digitally encoded representation of the visual characteristics of a scenery or the structure of an object or living thing [22]. Digital image has always been popular since its inception and it has been utilized in many applications. Digital image is commonly created by using digital cameras and scanners, but it can also be produced via drawing software or other techniques such as ray tracing. Once created, the image is then stored on a digital medium, which can then be edited with various image processing techniques, and finally be displayed either on a display monitor or surface such as a paper through printing.

With the advancement of technology, researchers strive to achieve higher image quality, which can not only increase the accuracy in various applications such as object tracking, forensic, medical, to name a few, but also for better viewing experience for pure entertainment purposes. Figure 1.0.1 shows five aspects in achieving high image quality, and a brief explanation on each aspect is summarized in Table 1.1. Specifically,

Table 1.1: Element of High Image Quality [3]

Factor	Explanation
Resolution	Fineness of Image Details
Bit depth	Fineness of Colour Gradations
Frame rate	Smoothness of Motion
Colour Gamut	Vividness of Colours
Brightness	Intensity of Image Lighting

resolution refers to the number of pixels in an image. With a fixed screen size, the higher the resolution, the more pixels there are, which allows finer details to be displayed. On the other hand, bit depth (colour depth) refers to the number of colours that can be displayed. The higher the bit depth, the more colours can be displayed, which can produce a smoother and more natural colour gradient. For example, an 8-bit display can show roughly 16.77 million unique colours, while a 10-bit display can show roughly 1.07 billion unique colours [3]. In the case of frame rate, it refers to the number of images shown within the period of one second. Films (analog) are generally shot at 24fps (24 images, or frames, per second), while standard TVs generally operate at 30fps (30 frames per second). Next, colour gamut refers to the range of realizable colours by the device and colours that are not representable in a given color space are called “out-of-gamut.” Last but not least, brightness is the range of the intensity values of the image that can be displayed. The difference between the darkest and the brightest objects (i.e., dynamic range) that the human eyes can perceive is empirically found to be 10^{12} , but traditional display devices are only able to display a range of 10^3 .

Conventionally, researchers focused on the development of high resolution devices for capturing and displaying. For example, the recently released smartphones (e.g., Samsung Galaxy S21 Ultra [23], Xiaomi Mi 11 [24], etc.) are equipped with 108 megapixels camera sensors as well as 8K resolution display. Thanks to their effort, consumers have these commercially available devices and therefore they can enjoy high quality media.

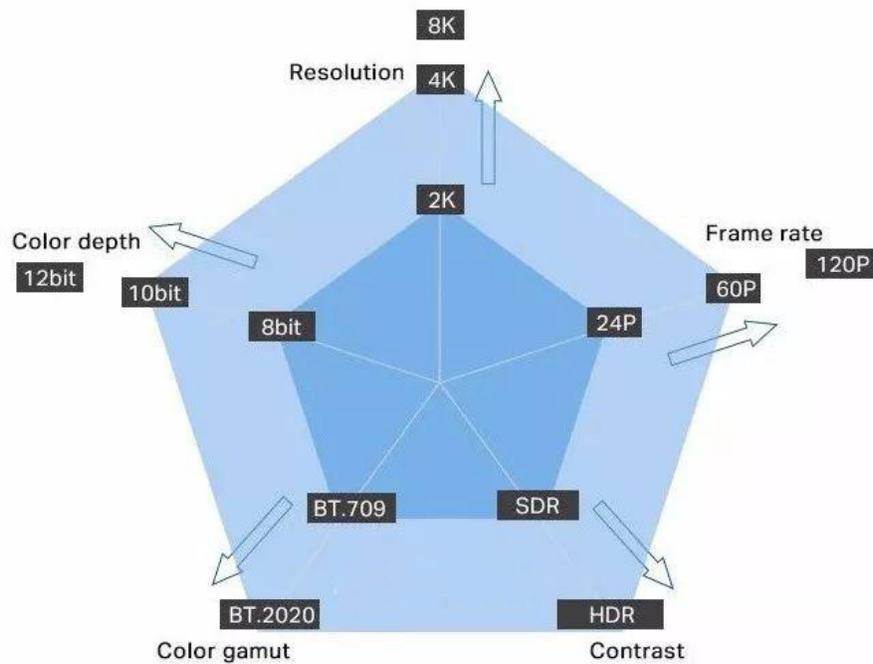


Figure 1.0.1: Five dimensions of image quality improvement [1]

While most effort for improving image quality is on producing image of higher resolution, there is also a trend in increasing the range of values that each pixel can represent. Recall that a majority of the colour images are represented with one byte (8 bits) per pixel per channel (i.e., red, green, and blue), which translates to 256 unique values per channel. However, the real-world scene is significantly more colourful. Increasing such range allows one to capture higher colour resolution and contrast to better capture the real world scene.

In order to digitally reproduce a scene much more similar to the real world as perceived by the human eyes, an imaging technique called *high dynamic range imaging* (HDRI) is created. High dynamic range (HDR) expands the dynamic range to 10^5 , which is a hundredfold increase over the traditional imaging technology. HDR allows details of the image (light and shadow) to be represented much closer to what the human eyes can perceive. The earliest applications of HDR include medical imaging,

astronomical photography and spectrographic research, which can be traced back to the mid-1960s. HDRI offers a new approach to represent colours in digital image. Instead of using the range of colours produced by a given display device, HDRI methods manipulate and store all colours and brightness levels visible to the human eyes. In order to store the digital data generated from HDRI, new file formats, which can store more than a byte per pixel per colour channel, are standardized specifically for HDR images. In addition, new display device (i.e, television and computer monitor) which can cater for the new technology are being designed and released to the consumer market by many major electronics companies.

Nowadays, with the advent of broadband Internet service, various types of digital contents can be easily shared among peers. Among the contents communicated, digital image is the most popular one because image can be easily captured with a smartphone/smart device and it can provide a self-explanatory message to the viewer. Since HDR image file formats are still new and gaining popularity, methods for protecting these images are scarce in comparison to the available methods for the traditional images. Some of the risks in sending digital images through the Internet includes unauthorized viewing, downloading and distributing the images. Therefore, in order to counter these wrongdoings, it is important to develop techniques to secure the HDR images. One of the ways to protect digital content is data hiding. Data hiding is the act of inserting data into a digital medium (i.e., image) by manipulating its content to serve specific purpose [25].

In digital image, data hiding is usually fine-tuned to specific purposes, including management of content via general embedding and protection of content via robust/fragile watermarking. Another problem which rises along with the invention of HDR imaging is the display of HDR image. Traditional capturing and display devices which are designed for standard dynamic range imaging are still in use by most part of

the society, partly due to the high cost for the newly invented HDR devices. Furthermore, when displaying traditional image in an HDR device, the hardware capability of a HDR display is not fully exploited. Therefore, tone mapping operator (TMO) and inverse tone mapping operator (ITMO) are designed to convert a HDR image to a standard dynamic range image, and vice versa. Scalable coding has also been introduced to achieve backward compatibility of HDR images. For the problem of mismatch in dynamic ranges (e.g., HDR image on SDR monitor), data hiding can also be employed for proper display and achieve scalable coding.

1.1 Problem statement

As introduced above, the rise of HDR imaging leads to security issues. Due to the advancement of the Internet, sharing and storing images using web services have become part of our daily life. Under such circumstances, the images uploaded online may be used by others without the owner's consent or they could be tampered with to stir up emotion or anger. Therefore, it is imperative to have methods to protect these images. One of the solutions is data hiding. In addition to the conventional applications of data hiding, reversible data hiding (RDH) in HDR image also has unique applications such as storing instructions / parameters for tone-mapping, as well as encoding a second impression of the tone-mapped low dynamic range (LDR) image produced by using a specific tone-mapping operator, which differs from the common operator, such as Reinhard's TMO [26]. However, research on data hiding in HDR images is still lagging in contrast to the advancement of HDR imaging. Therefore, in this research, data hiding techniques will be investigated for HDR images. Here, both aspects of data hiding, i.e., data embedding and image masking, will be studied.

In addition to addressing the security needs in managing HDR image, another im-

portant problem to be mitigated is backward-compatibility. Specifically, although HDR display devices are available in the market, most display devices in used nowadays can only handle LDR media. This is because HDR display devices are still new and they are expensive for the most consumers. As one would expect, LDR display device cannot display HDR image correctly due to the lack of brightness producible by the existing hardware as well as the correct graphic card to render the image.

Therefore, scalable coding is needed to allow backward compatibility and adaptation to future display technologies. Specifically, scalable coding of HDR image will be investigated to provide a progression quality improvement decoding mechanism in terms of bit depth.

1.2 Research questions

This research aims to answer the following questions:

1. What characteristic of HDR image can be exploited to hide data?
2. What characteristics of HDR image can be exploited for image masking?
3. How to achieve backward compatibility and adaptation to new display devices using scalable coding?

1.3 Research objectives

To answer the research questions stipulated above, this study embarks on achieving the following objectives:

RO1: To design an efficient data hiding technique for HDR image. The goal is to be able to embed external data into a HDR image while maintaining high quality. High

quality output image is required to ensure the imperceptibility of the embedded data.

RO2: To design an image masking technique for HDR image. The main task here is to transform an HDR image into a perceptually unrecognisable form while allowing the reconstruction of the original image.

RO3: To design and implement scalable coding technique for HDR image to complement the existing file format/standard in order to achieve backward compatibility. The main purpose here is to transcode a HDR image bit stream for allowing direct decoding using existing decoders.

1.4 Scope of project

This study first investigates into the characteristics of HDR in general, then it focuses on a specific file format - i.e., OpenEXR, to demonstrate the feasibility of achieving data hiding, image masking, and scalable coding. The OpenEXR format is considered because it is the most popular file format in the industry. In addition, the HDR images from the Fairchild data set [21] are utilized for experiments and analysis purposes in this research. The Fairchild data set is utilized because the images are mainly made up of outdoor scenery, which gives a good representation of natural images. Analysis are then performed to measure the perceived quality of the processed images (i.e., data-embedded, or masked-and-data-embedded) as well as the capacity available for data hiding. For the case of reversible data hiding, the main goal is to ensure that the original image can be reconstructed perfectly from the processed image. It should be noted that the processed image (embedded with data) can be of lower quality than the original image but the processes carried out to embed data can be ‘reversed’ to completely restore the original image. The framework in achieving reversible data

hiding will be completed and presented in the thesis.

Next, the combination of data hiding and image masking in the OpenEXR file format is investigated. It is expected that the proposed method can achieve image masking and at the same time contains hidden information. Unlike the first method, this method aims to distort the quality of image while inserting data.

Finally, scalable coding technique is studied for HDR images. With the designed technique, it is expected that HDR image can be decoded to different bit depth to suit the display device in use. For example, the image can be decoded to a bit depth of 8-bit per channel when an older device is used so that the resulting image contains a little visible artifact or distortion as possible.

It is beyond the scope of this project to investigate into how the tone mapping operators work. Similarly, it is acknowledged that other HDR file formats are also in use, but for the purpose of this work, the focus is on the OpenEXR file format.

1.5 Structure of the thesis

This thesis consists of six chapters. Chapter 1 gives a brief introduction on HDR images and the importance of data hiding for them. In Chapter 2, HDR image, HDR tone mapping and common data hiding and image masking methods for traditional images as well as recent proposed methods for HDR image are surveyed. After that, two reversible data hiding methods for HDR image are proposed in Chapter 3. Chapter 4 proposes an image masking and data hiding method for HDR image. In Chapter 5, a scalable coding method for HDR image is proposed. Finally in Chapter 6, the conclusion and suggested future work are presented. Figure 1.5.1 shows the overall view of each topic of the thesis.

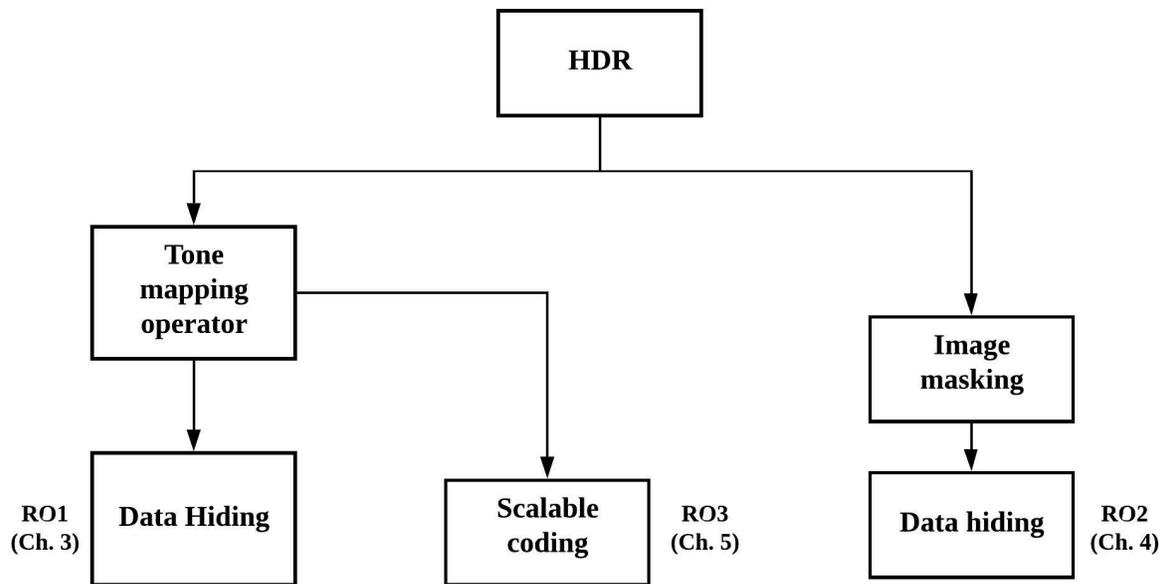


Figure 1.5.1: Overall structure of the thesis

Chapter 2

Literature review

This chapter first introduces the characteristic of human visual system as well as the importance of luminance and dynamic range in recreating real world scene as seen by human eyes. Then, an overview on high dynamic range imaging as well as various file formats are provided. Subsequently, techniques to protect digital image, data hiding and image masking, are review for both traditional and HDR images. Finally, methods for ensuring proper display of HDR image in traditional display device such as tone mapping operator and scalable coding are also surveyed.

The goal of this chapter is to provide a holistic overview of the HDR imaging technology by describing the display technology, image acquisition techniques and HDR file formats. There can be two levels of readings, where experts can leave out some parts, which are useful to non-specialists for a good understanding of HDR image technology. Experts on HDR imaging can start reading from Section 2.6.

2.1 Human visual system

The eye is an organ that gathers light onto photoreceptors, which then convert light into signals [2] as shown in Figure 2.1.1. Specifically, light enters the eye and first passes

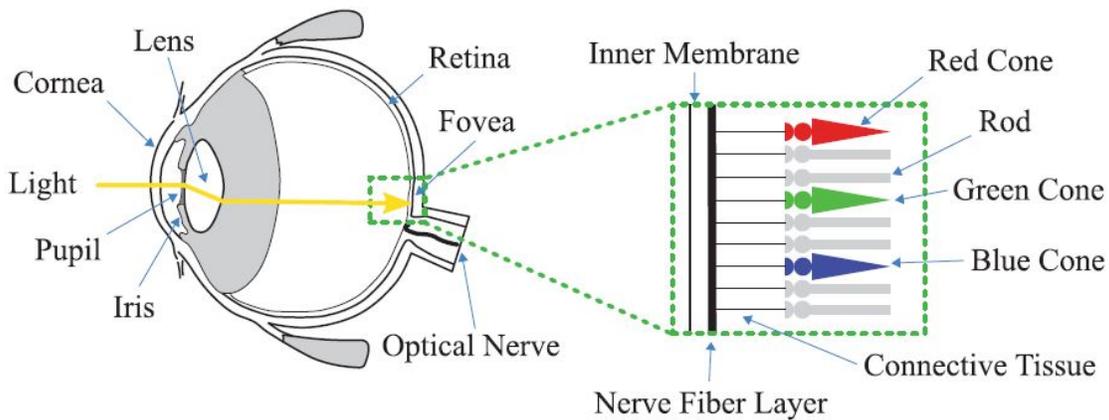


Figure 2.1.1: The human eye [2]

through a transparent membrane called cornea. Then, the light enters the pupil, which is an aperture that is controlled by the iris. Finally, light is refracted by the lens and it hits the photoreceptors in the retina. The received signals are then transmitted to the visual cortex through the optical nerve. The visual cortex is an area of the brain that processes these signals to produce the perceived image. This full system of human body is responsible for vision and is referred to as the human visual system (HVS) [2].

There are two types of photoreceptors, namely, cones and rods. The cones are sensitive during daylight, and they are responsible for the perception of high frequency patterns, fast motion, and colours. On the other hand, the rods are for nightvision and they are more sensitive than cones but do not provide colour vision. The ratio of rods and cones is approximately 20 : 1 [27], and this is the reason why human is unable to differentiate between colours and cannot distinguished high frequency patterns under low level illumination conditions. Due to this characteristic of the human visual system, luminance has been identified as an important factor in recreating real world scene as seen by human eyes through digital device. As illustrated in Figure 2.1.2, the human eyes can usually perceive luminance ranging from 10^{-6} to 10^6 cd/m^2 (Candela per square metre, derived SI unit of luminance), i.e., range of $\sim 10^{12}$. High dynamic range

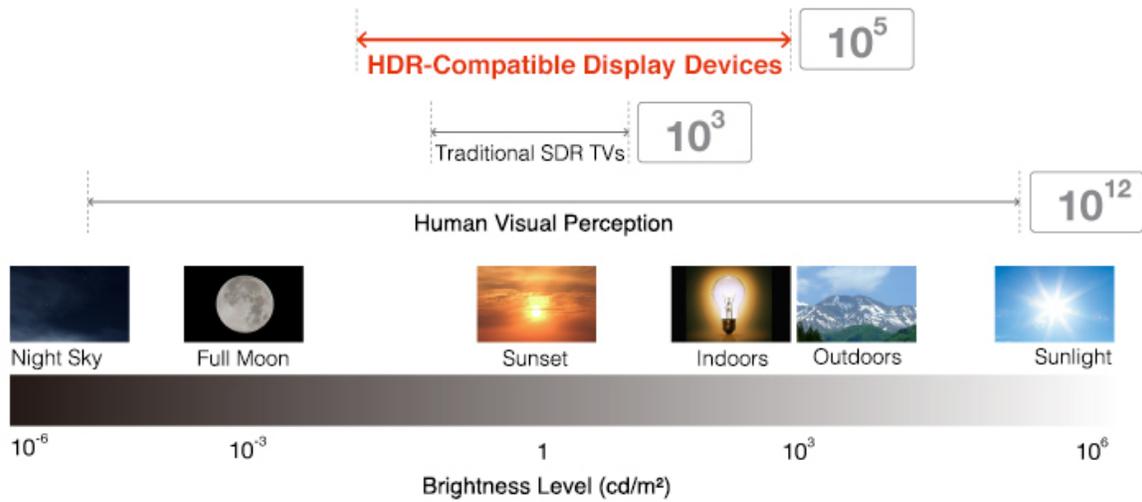
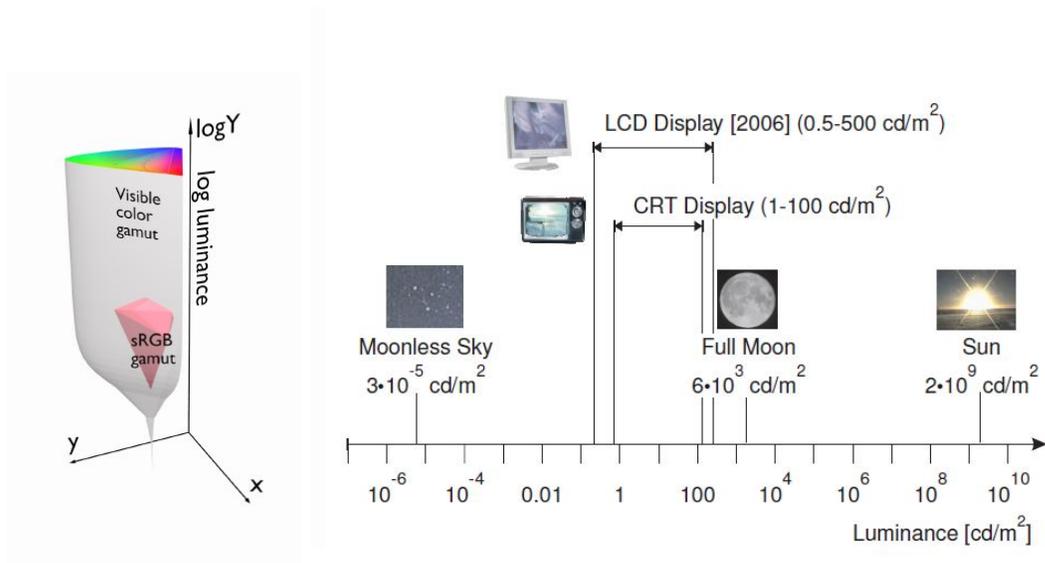


Figure 2.1.2: Dynamic range [3]

imaging can capture the luminance value with much higher accuracy in comparison to the standard dynamic range and therefore it can outperform the traditional imaging method.

2.2 High dynamic range imaging

Most existing digital imagery can only capture, at one particular instance, a portion of the visual information that is visible to the human visual system, and hence they are incapable of reproducing the actual scene entirely. Since most recent consumer-level digital cameras can take images of higher number of pixels than most of displays can offer, resolution is not the limiting factor in regenerating the scene. However, the main problem in generating a high quality image is the limited colour gamut and even more limited dynamic range captured by cameras and the representation used in most image file formats. Therefore, the images generated by traditional imaging technology are called low dynamic range (LDR) to highlight these limitations [4]. LDR has also been referred to as standard dynamic range (SDR) in the literature, but for the purpose of



(a) Visible colour gamut

(b) Luminance range

Figure 2.2.1: Visible colour gamut and luminance range [4].

this thesis, the term LDR will be used. Here, dynamic range (DR) refers to the highest overall contrast that can be found in an image [28].

To date, most colour images are stored in LDR and they are represented as an assemble of 24-bit pixels, where each pixel contains the red, green, and blue channels each represented by using 8 bits. With three bytes per pixel, about 16.7 million different colours can be assigned to each pixel. This is known as true colour or “millions of colours”. This may seem a tremendous number, but it should be noted that there are still only 256 values for each of the red, green, and blue channel of each pixel [13]. Since the range of the displayable luminance is often capped in LDR system (i.e., file format and display device), this phenomenon leads to the differences between the acquired digital image and the view by the human eyes.

Consequently, with only an 8-bit integer per channel, only a small part of visible colour gamut can be represented and an even smaller part of the luminance range

that can be perceived by human eyes. Figure 2.2.1(a) shows the visible colour gamut by the human eyes and Figure 2.2.1(b) the visible luminance range. The reason for adopting this limited range of integers is because when digital image was just getting popular, most image format was designed to store as much information as what can be displayed on the majority of displays, which were at that time the cathode ray tube (CRT) monitors or TV sets. However, this limitation is no longer valid. New generations of display devices such as LCD, plasma, LED and OLED displays can portray a much wider colour gamut and dynamic range than the legacy CRT monitors. The new generation displays can offer better colour reproduction and are capable of displaying higher precision of digital image and video content. Another problem in three 8-bit integer colour channels is the low contrast range and hence limited colour gamut, which cannot offer the precision required for the advancement in image capturing, processing, storage, and display technologies.

To overcome these problems, high dynamic range imaging (HDRI) is introduced. HDRI can overcome the limitations of traditional imaging due to its higher precision in colour data. In particular, HDR image is stored as an ensemble of triplets of floating point values, which can provide higher accuracy and represent more colours in real world perceived by the human eyes. [29]. High dynamic range (HDR) imaging, as its name implies, is able to capture and display significantly more colours when compared to the legacy imaging technologies. Figure 2.2.2 shows the HDR imaging pipeline, starting from image acquisition to reproduction.

2.2.1 HDR image acquisition

There are a number of ways to capture HDR contents. Due to the costly HDR capturing hardware, most HDR images in circulation nowadays are generated with the aid of software. Specifically, consumer cameras can only capture 8-bit or 14-bit images in

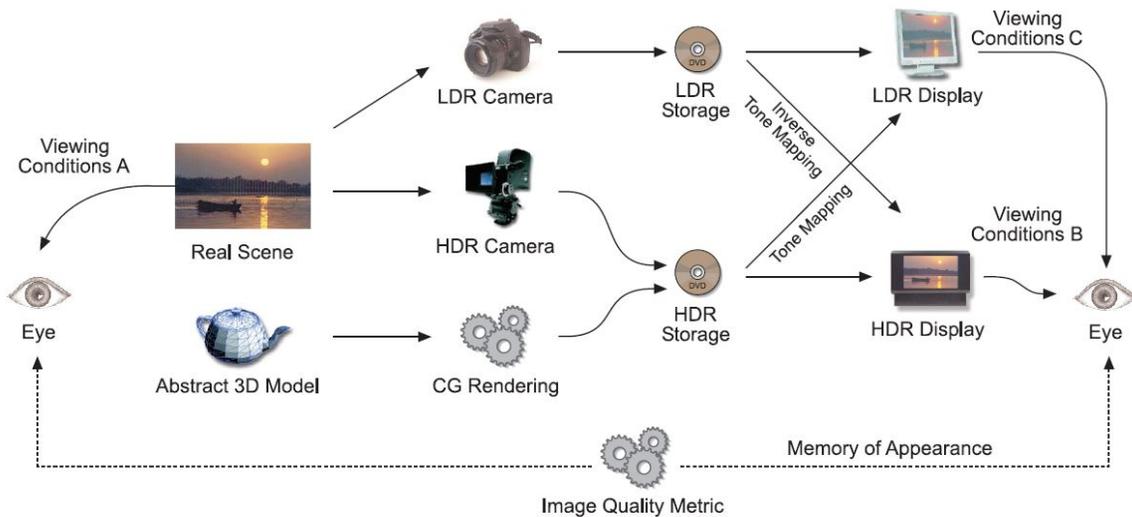


Figure 2.2.2: HDR imaging pipeline, starting from image acquisition to reproduction [4]

RAW format and do not cover the full dynamic range of luminance values in real world environments. Therefore, the most common method to obtain HDR image is to capture multiple images of the same scene at different exposures to capture the details from the darkest to the brightest areas. This set of images is called an exposure stack [2]. Subsequently, the HDR image can be constructed by blending this stack of images. An example of exposure stack and generated HDR image is shown in Figure 2.2.3 and Figure 2.2.4.

2.2.2 Representation of HDR image pixels

A HDR image is usually stored as a rectangular matrix of floating point values (viz., pixels). Unlike the standard dynamic range where the step size are all equal, HDR imaging usually adopts the logarithmic step size to represent the dynamic range of a scene. The dynamic range is calculated as the difference between the logarithm (base 10) of the brightest and the darkest spots. Considering this large range of values, in-

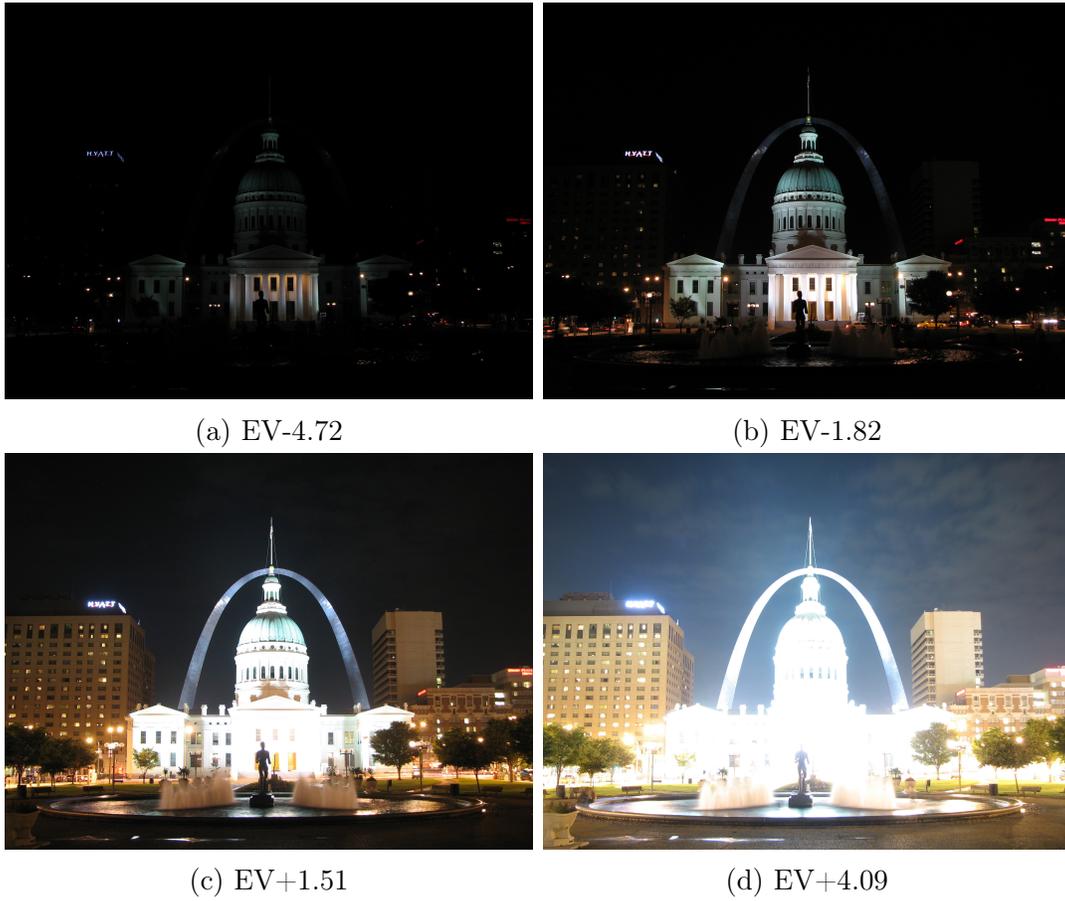


Figure 2.2.3: An exposure stack of images captured at different exposure levels [5].



Figure 2.2.4: The HDR image resulting from multiple exposure blending. The HDR image is tonemapped for display purposes [5].

teger values used in SDR (or LDR) imaging are inefficient to store HDR values. For instance, a 32-bit signed integer has a maximum value of $2^{31} - 1$ which is insufficient to cover the entire range perceivable by the HVS. On the other hand, the 32-bit floating point representation can have a maximum value of $(2 - 2^{-23}) \times 2^{127}$, which is a value significantly larger than the maximum value that can be captured by a 32-bit integer representation. In other words, although both representations have 2^{32} unique steps, the floating point representation values can be mapped to a larger dynamic range. Furthermore, integer values are also unsuitable for image processing operations between two or more HDR images. For example, precision is easily sacrificed or overflow may occur when performing different arithmetic operations such as adding or multiplying. Based on these characteristics, it can be justified that floating point values are more suitable, in comparison to integer, to capture the real-world values [2, 30]. The adoption of logarithmic functions ensure better representation of the perceived difference in dynamic range, as it is similar to the human visual system in its reaction to light. The logarithmic function of HDR makes the floating point value a more suitable data type for storing the pixels. Furthermore, floating point can cater for a wider range of numbers with fewer number of bits in comparison to integer data type.

2.2.3 HDR file formats

By construction, HDR content will typically have larger file size than its SDR (LDR) counterpart due to the nature of values stored (i.e., floating point values vs integers). To cater for the pixel representation of HDR image in floating point format and to make it more manageable, new pixel encoding techniques and file formats have been designed, including JPEG2000, JPEG-XR, JPEG-XT [31], OpenEXR [32], Radiance RGBE [33] and LogLUV TIFF [34].



Figure 2.2.5: *RGBE encoding*

Radiance RGBE

Radiance RGBE was first introduced by Greg Ward in 1991 [33]. The main purpose of this file format is to reduce the number of bits by using one 8-bit mantissa for each colour channel and a shared 8-bit exponent. Similar to most floating point formats, the mantissa is normalized to stay between 0.5 and 1 [33]. However, due to the shared exponent for three mantissas, only the largest value is guaranteed in this normalization, and the other two may be less than 0.5. The floating point representation for RGBE file format is shown in Figure 2.2.5. Each of the R , G , B and E components can be computed from the floating point values as follows [35]:

$$R = \left\lfloor \frac{256R_W}{2^{E-128}} \right\rfloor, \quad (2.1)$$

$$G = \left\lfloor \frac{256G_W}{2^{E-128}} \right\rfloor, \quad (2.2)$$

$$B = \left\lfloor \frac{256B_W}{2^{E-128}} \right\rfloor, \quad (2.3)$$

and

$$E = \lceil \log_2(\max(R_W, G_W, B_W)) + 128 \rceil, \quad (2.4)$$

where W refers to world, i.e., the original values captured from real world scene. At the decoding stage, the RGB and E values are converted back to R_W , G_W and B_W as



Figure 2.2.6: 32-bit LogLUV bit distribution

follows:

$$R_W = \frac{R + 0.5}{256} (2^{E-128}), \quad (2.5)$$

$$G_W = \frac{G + 0.5}{256} (2^{E-128}), \text{ and} \quad (2.6)$$

$$B_W = \frac{B + 0.5}{256} (2^{E-128}). \quad (2.7)$$

LogLuv TIFF

In [34], Ward et al. proposed a new image encoding scheme called LogLuv encoding, which uses a logarithmic representation for luminance and the CIELUV representation for chrominance. CIELUV is a simple-to-compute colour space transformed from CIE XYZ, a colour space derived from RGB colour space. This format covers the entire visible colour gamut and the full range of perceivable luminance. It also uses imperceptible step size in a perceptually uniform space. Specifically, 32-bit LogLuv encoding assigns 16 bits for luminance L and another 16 bits for both chrominance channels u and v . In addition, the MSB is reserved to flag a negative or positive luminance. To convert a pixel value into the LogLuv format, the RGB colour space is first converted to into the XYZ. Then the XYZ value can be converted to L_e , u_e and v_e by using the equations below:

$$L_e = \lfloor 256(\log_2 Y + 64) \rfloor, \quad (2.8)$$

$$u_e = \lfloor 410u' \rfloor, \quad (2.9)$$

$$v_e = \lfloor 410v' \rfloor, \quad (2.10)$$

$$u' = \frac{4x}{-2x + 12y + 3}, \quad (2.11)$$

$$v' = \frac{9y}{-2x + 12y + 3}, \quad (2.12)$$

$$x = \frac{X}{X + Y + Z}, \text{ and} \quad (2.13)$$

$$y = \frac{Y}{X + Y + Z}, \quad (2.14)$$

$$(2.15)$$

A scale factor of 410 is chosen here to ensure the integer generated lays between 0 to 255, as the gamut of perceivable u and v lay between 0 and 0.62.

The main advantage of LogLuv is that the format stores luminance and chrominances information separately, making the file more practical for applications such as tone mapping.

OpenEXR

OpenEXR is an open-source HDR image file format developed by Industrial Light & Magic [32]. It can support a higher dynamic range and better colour precision than the traditional 8-bit file format. OpenEXR file is capable of storing image based on floating point. Each colour component is of 16-bit or 32-bit floating-point data type, and the pixels can be compressed in either lossless or lossy manner. It can have a number of user-defined channels, each with a different data type. The most common channels are the R, G and B channels. [32]

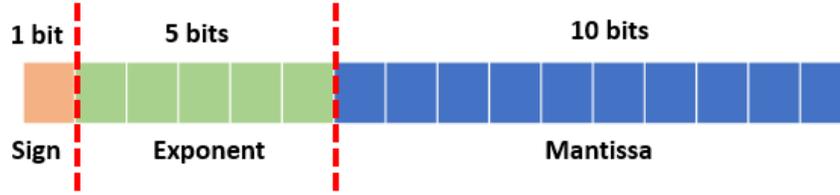


Figure 2.2.7: *Illustration of a 16-bit pixel in the OpenEXR half-precision file format. Reproduced from [6]*

Table 2.1: Semantic of 16-bit floating point representation in OpenEXR.

Condition	Value
$E = 11111_2 = 31$ and $M > 0$	NaN
$E = 11111_2 = 31$ and $M = 0$	\pm Inf
$E = 0$ and $M > 0$	Subnormal Number
others	Normalized Number

The 16-bit half format, which is also called the ‘half-precision’ format, is a simplified version of the IEEE 754 floating-point [36] specification. Specifically, half precision consists of 1 sign bit (denoted by S), 5 exponent bits (denoted by E), and 10 mantissa bits (denoted by M). Figure 2.2.7 shows the semantic of each group of bits. The half format supports denormalized numbers, positive and negative infinities, and NaN (i.e., not a number). The range of positive representable values of half format falls between $2^{-24} \approx 5.96 \times 10^{-8}$ and $(2 - 2^{-10}) \times 2^{15} = 65504$. Specifically, the value denote NaN if $E = 11111_2 = 31$ and $M > 0$. If $E = 11111_2 = 31$ and $M = 0$, the values denotes infinity, where the sign is determined by S . When $E = 0$ and $M > 0$, the value represented is a subnormal number, which is a non-zero value less than the smallest normal number. Other than the aforementioned combinations of E and M , the value represented is a normalized number, with a hidden leading 1. Table 2.1 summarizes all types of representable value in OpenEXR format.

JPEG-XT

Joint Photographic Experts Group (JPEG) finalized a new standard called JPEG-XT in year 2015 [7]. In practice, JPEG-XT is backward compatible to the legacy JPEG [37]. The main idea of the JPEG-XT coding standard is to utilize 2 or more layers to store a HDR image. First, the HDR is tone-mapped to SDR, where the resulting SDR image is encoded by using the legacy baseline JPEG compression standard. This form the base layer in the JPEG-XT image. The JPEG image from the base layer is decoded and inversely mapped to a wider dynamic range, and the difference of this new HDR image and the original HDR image is computed and stored in the enhancement (residual) layer. In other words, JPEG-XT encoder handles a pair of images: an HDR and an LDR image where the LDR image is a tone mapped version of the HDR image, which is stored in the legacy JPEG format [7,31,37]. In fact, JPEG and JPEG-XT appear to be the same format, except that the later has more information in the application marker (i.e., APP11). For detailed information about the JPEG-XT compression standard, interested reader may refer to [31,38].

Hence, the users can decode LDR image from a JPEG-XT bitstream via any existing JPEG decoder. On the other hand, when using a JPEG-XT decoder, the application maker APP11 is located, and the information informed by the marker is referred to decode the residual image. The base layer (i.e., LDR) image and the residual layer (i.e., HDR) image is combined to found the final HDR image. Figure 2.2.8 shows the JPEG-XT decoder as defined in [7].

2.2.4 File format chosen

In this research, the OpenEXR file format is chosen as the file format to be studied. This is because OpenEXR is the most popular file format in the industry nowadays. The

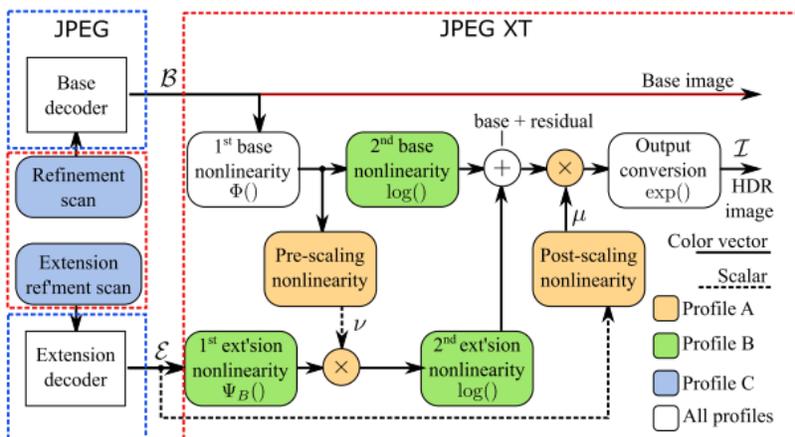


Figure 2.2.8: *JPEG-XT decoder* [7]

OpenEXR file format was originally developed by Industrial Light & Magic (ILM) and it was released in 2003. Weta Digital, Walt Disney Animation Studios, Sony Pictures Imageworks, Pixar Animation Studios, DreamWorks, and other studios, companies, and individuals have made significant contributions to the code base [39]. Since its introduction, OpenEXR is used in popular films such as *Harry Potter and the Sorcerer’s Stone*, *Men in Black II*, *Avengers: Infinity War* and *Star Wars: The Last Jedi*, to name a few [40].

2.3 Display devices

As introduced earlier, a digital image is an assemble of pixel values in the form of a rectangular matrix. Specifically, pixel values describe both tones and colours of the image. When more bits are assigned to each pixel, more unique colours can be represented, and vice versa. Since computers usually operate on data in the unit of byte (viz., 8-bit), this translates to 256 levels of gray tone or 256 unique colours.

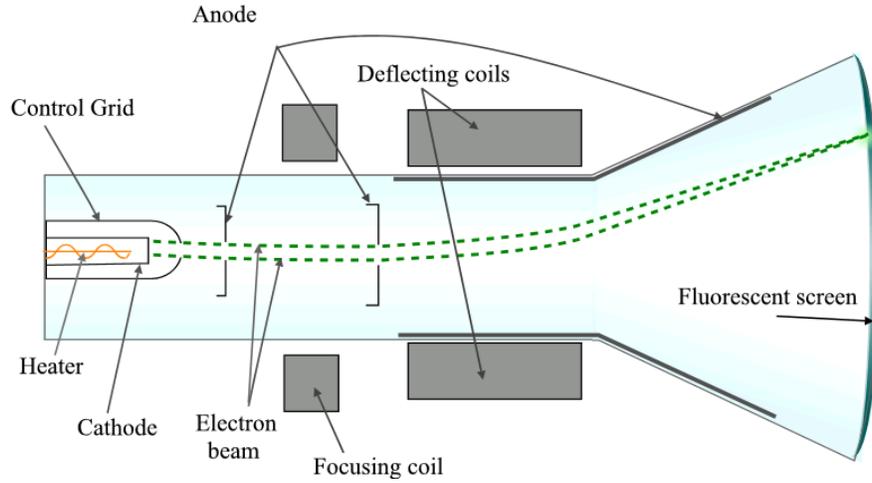
To reiterate, the range of colours are represented by using three primary colours, namely, red (R), green (G) and blue (B). An image rendered on a display device can

trick the human eyes into seeing colours with different light intensity generated by the three colours. In other words, the colours perceived by the human eyes are actually a mix of three primary colours. To display colour on a monitor, firstly, the image file containing the grid of pixel values is obtained. Essentially, each pixel is a triplet, which represents the intensity level of each of the R, G, and B colour channels. The monitor then converts each colour value into lights of red, green and blue, with intensities as stipulated by the stored values. Legacy display devices have 8-bit per channel, which can have 256 variations of colour for each channel. By combining the three channels, about 16.7 million colours can be formed. In the following subsections, four types of display devices are discussed.

2.3.1 Cathode Ray Tube display

The cathode ray tube (CRT) is a vacuum tube in the device configured to control and focus an electron beam to strike a phosphor-coated surface that converts the energy into light. The intensity of light generated on the display surface (screen) depends on the amount of electricity the electron gun shoots, where the more electrons the brighter the colour is. There are three types of phosphor: red, green and blue, all arranged in a tight matrix on a display surface and a pixel is portrayed by each of these phosphors. Colours seen on the display surface are the combinations of these three colours. To ensure continuous lightning of the phosphor, CRT scans the display surface at a high frequency, fooling the human eyes into perceiving the continuous presence of a 2D colour image.

A CRT display typically receives analog signals through VGA interface, but some CRT display are equipped with digital-to-analog (D/A) converter to incorporate the Digital Visual Interface (DVI) interface.



(a) Cathode ray tube [41]



(b) CRT monitor [42]

Figure 2.3.1: CRT display device

2.3.2 Liquid Crystal Display

For liquid crystal display (LCD), the basic theory of displaying pixels is almost the same as CRT display, that is, the combination of red, green and blue lights are utilized in tricking human eyes into perceiving colours. Specifically, LCD consists of liquid crystal film and polarized glass (see Figure 2.3.2). The first layer of polarized glass only allows horizontal light from the backlight panel to pass through. Then, the liquid crystal can control the direction and angle of light passing through from the first polarized

glass by using electric current supplied to the electrode. This is possible because the molecular structure of liquid crystal will change when electric current passes through it. Subsequently, the second polarized glass can filter the light in wrong direction (i.e., light not in horizontal) to ensure a better viewing experience for user.

Since LCD does not contains the bulky electron gun, the LCD display tends to be flatter and thinner in comparison to CRT display. Unlike CRT display which is continuously striking a phosphor to keep pixel lighting, each pixel in an active matrix LCD is held constant by the combination of a capacitor and a transistor, which acts as a short-term memory circuit between refreshes. LCD is capable of producing higher luminosity but it is technical challenging to produce an absolute dark state as the back light will still leak through the polarized glass even when no current is passing through the liquid crystal [43].

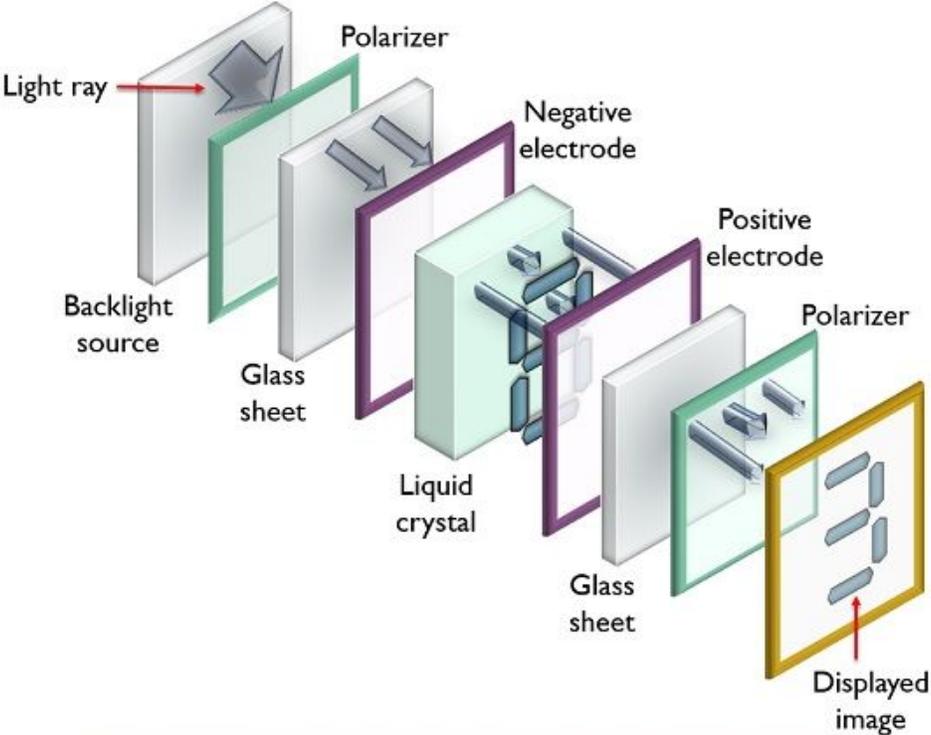


Figure 2.3.2: Working structure of LCD [8]

2.3.3 Plasma display

Similar to CRT, plasma display excites phosphor to render colour, but unlike CRT, plasma display uses ultraviolet photons released from collision of electrically ionized gas to hit the phosphor of different colours. Gas atoms are stored in cells and ionized by electrodes. When the electric current passes through the electrode, electrons are introduced into the gas atoms and subsequently collision occurs. Gas atoms then turn to ions quickly and the collision of ions leads to the emission of photons. These photons are responsible to light up the phosphor. Figure 2.3.3 shows the basic structure of a plasma display.

The advantage of plasma display is that extremely thin materials can be utilized to produce a wide screen. The pixels can be made brighter than CRT display, as radiation is not an issue in plasma display. Although the darker pixel is not as dark as rendered by the CRT display, the brighter pixels can contribute in producing high contrast display. However, since plasma screen are made of glass, it reflects more lights from the surrounding objects when viewing, i.e., glare. The plasma display typically consumes more energy than the LCD and CRT displays. It is also significantly heavier and more fragile than LCD, which causes difficulties in shipping the device. These disadvantages become the main reasons manufacturers stopped producing plasma display [44–46].

2.3.4 Light Emitting Diode display

The light emitting diode (LED) display is basically an LCD display with LED as the back light source instead of a fluorescent tube. LED is usually more energy efficient than fluorescent tube, as it consumes less power when lighting up. In addition, LED display is lighter, thinner and less bulky than LCD display. LED can give a wider colour gamut therefore it can display more vivid images. However, the LED display is

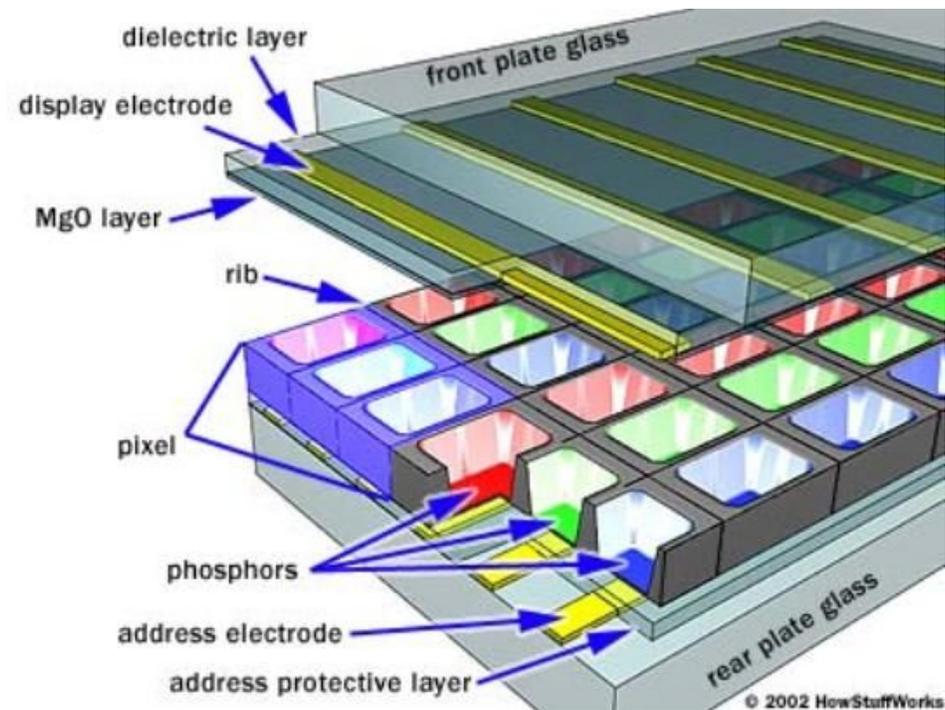


Figure 2.3.3: Structure of plasma display [9]

usually more costly than the LCD display. [47]

2.3.5 Organic Light Emitting Diode (OLED) display

The organic light emitting diode (OLED) display is a huge breakthrough of display technology. First appeared in year 2013, the device quickly attracts the attention of consumers. OLED is made by placing a series of organic thin films between two conductors (Figure 2.3.4). When electrical current is applied, a bright light is emitted [10]. Unlike LCD and LED back light display, OLED display does not require a back light as it can light up by itself, therefore it is thinner and more efficient. Since OLED is thin, flexible, and remarkably small, it can be used as individual pixels occupying the screen, which can light up and shut off independently. Therefore, OLED display uses less power, and makes total dark state possible. However, OLED display device is expensive as it is still a relatively new technology [48, 49].

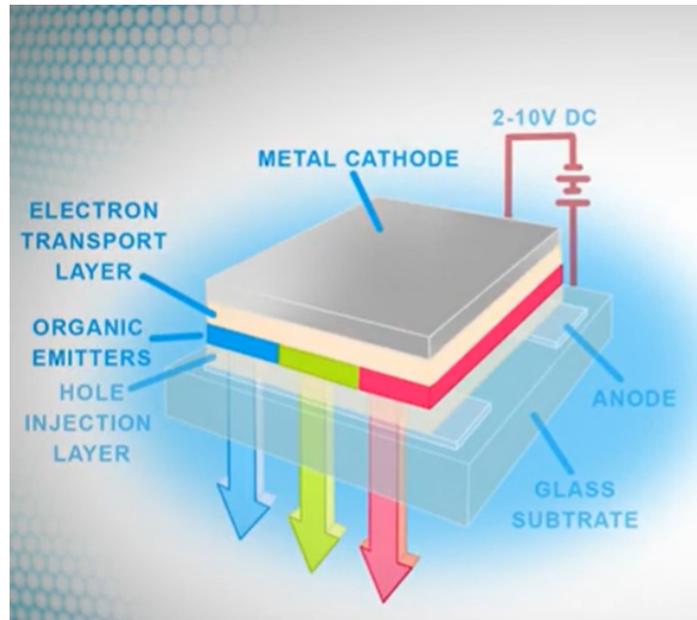


Figure 2.3.4: Structure of OLED. [10]

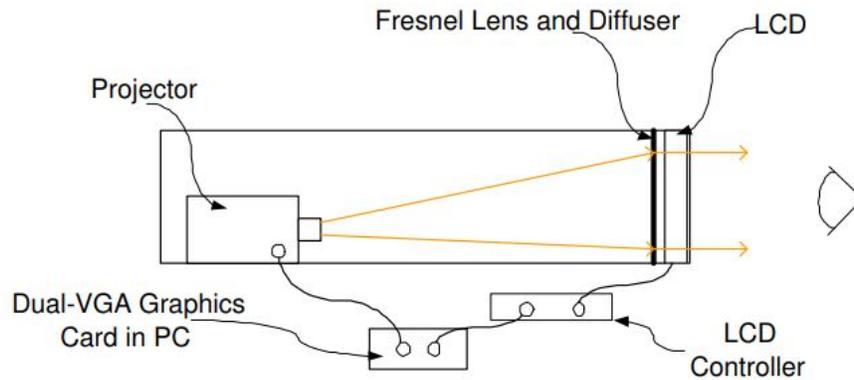
2.3.6 HDR display devices

HVS can differentiate up to ten million colours and greatly depend on the brightness due to the characteristic of the photoreceptors in the human eyes. Therefore, the HDR display devices that have been developed recently have specification which take in to account the trait of the HVS in terms of reproduced contrast and brightness levels. To replicate the real world scene in a display device, two basic approaches can be considered [4]:

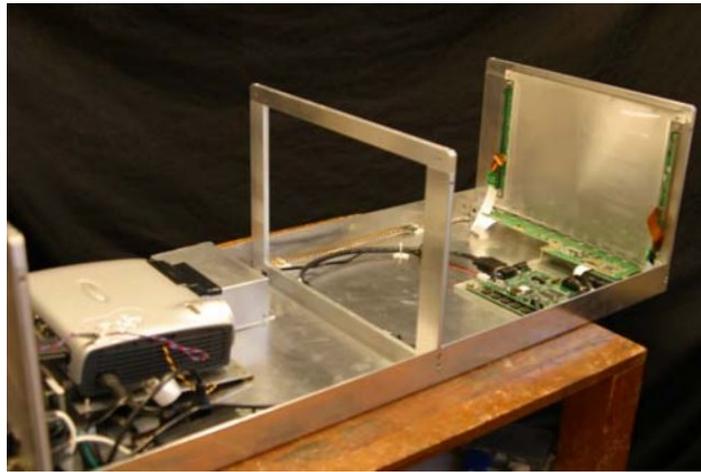
1. A direct precise modulation of each pixel over a very wide luminance range, and;
2. The serial combination of two or more modulators to achieve the same effect.

The first approach is more difficult to achieve technologically because 12-16 bit-depth precision is needed to display each pixel. When displaying, zero luminance (total dark pixel) and a high luminance value ($3,000 - 6,000 \text{ cd/m}^2$) should constantly be available and at the same time, no significant light leak between neighboring pixels should occur.

The second approach is technologically more feasible and achievable. In constructing HDR display with dual modulation, the rendering process relies on optical combination of two independently modulated representations of the same image and the resulting image contrast is a product of contrast achieved by each component image. In this case, for each modulator only the standard 8-bit drivers are required to control the pixel values. For rendering, the first modulator (the backlight) actively emits controllable amount of light similar to the grid of LEDs. The second modulator, which is illuminated by the backlight, is a passive transmissive LCD panel that controls the amount of per-pixel transmitted light [4]. The basic design of the first dual modulation HDR display is proposed in [11] by Seetzen *et al.*. In the proposed design, HDR display is achieved by using a projector as a backlight (i.e., first modulator), and a LCD as the second modulator. Figure 2.3.5 shows the set up of the proposed display, and Figure 2.3.6 shows the overall rendering process in the dual-modulation display architecture. However, this proposed display is impractical due to several reasons. The required optical length for the projector is long, which is not practical for home entertainment. Furthermore, the cost of a high brightness projector is very high which makes this proposed version of the HDR display not suitable for commercial purposes. Finally, the utilization of very bright projector needs high power consumption and will generate significant amount of heat. Due to these weaknesses, Steetzen *et al.* proposed another design which uses LED as the backlight, replacing the projector. The utilization of LED eliminates the problem of high power consumption and high heat dissipation, hence it is more suitable to be commercialized. This method of dual modulation using LED and LCD becomes the basis of many later design such as the Brightside Technologies' DR37-P (Figure 2.3.7) [12,50], the world first HDR display. Today, HDR display technology even has its own certification (i.e., VESA DisplayHDR [51]) to give consumers information about the display capability of a device.



(a) Schematic diagram.



(b) Actual display.

Figure 2.3.5: Structure of first proposed HDR display. [11]

2.4 HDR image quality assessment

The wider dynamic range of HDR image makes it unique when compared to the traditional LDR image. This feature of HDR image allows us to better reproduce the real world scenes with the aid of advanced display technologies. Similar to any technology, it is important to evaluate HDR systems in terms of quality. Older quality metrics which are not designed for wider range of luminances and do not take into consideration about the sensitivity of human visual system towards luminance are no longer suitable to determine the quality of HDR image.

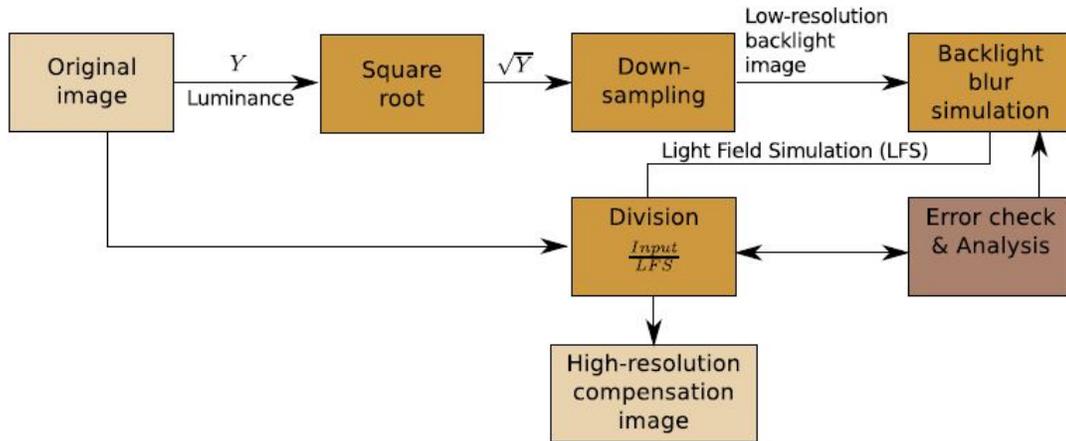


Figure 2.3.6: The flow of processes for driving the low-resolution backlight modulator and the high-resolution front LCD panel in HDR display [4, 11]



Figure 2.3.7: World first HDR display, DR37-P [12].

Human observers are very good in immediate judgment of image quality, even without any reference images. Hence, subjective image quality evaluation with human subjects remains the most reliable approach. However, individual factors such as preferences, expectations, and past experiences can affect evaluation process. To reduce influence of such factors, the quality rating based on human evaluation is usually averaged in the form of mean opinion score. Furthermore, the influence can also be reduced by a well-planned experimental procedure, precise instructions, and pilot sessions offered to the subjects [13].

Nevertheless, subjective methods are costly and time-consuming, where the subjects are required to have the knowledge and they need to put in effort in producing meaningful quality scores. Due to this, many applications prefer to adopt objective assessment, which will not be affected by human factors. Besides, objective methods are the better choice in applications dealing with a huge number of images, which may be tedious for subjects to score each of them. Objective image quality metrics are often classified with respect to the amount of information required from the reference image to evaluate the distorted image quality. There are three categories, namely:

- full-reference, where the reference (unprocessed) image is assumed to exist and fully available;
- reduced-reference, where certain features are extracted from the reference image and used as side information, and;
- no-reference, where the process is blind without any reference [52].

Based on [53], only three reliable objective metrics have been specifically designed for HDR content, including: (a) HDR visible difference predictor (HDR-VDP) metric proposed by Mantiuk et al. [54]; (b) HDR-VDP-2 [55], which is the improved version of HDR-VDP, and; (c) the recently proposed high dynamic range video quality metric

(HDR-VQM) by Narwaria *et al.* [56]. HDR-VDP-2 is considered as the current state-of-the-art assessment metric for HDR image quality evaluation. Furthermore, due to the existence of different display devices with different specifications, it is important to take into account these differences when measuring the quality. Such metric is referred to as the display-referred metric. Metrics which take in any relative HDR pixel values and give identical results are called luminance-independent metrics [4].

2.4.1 Luminance-independent metrics

Luminance-independent metrics usually convert HDR pixel values into the logarithmic domain. This is because, based on Weber’s law, human visual sensitivity towards luminance is approximately similar to a logarithm function [57]. An example of such a metric is $\log PSNR$ (peak signal-to-noise ratio), which is the modification of the standard PSNR formula. Specifically, the $\log PSNR$ metric is computed with the logarithmic expression as shown in Eq. (2.16) [4]:

$$\log PSNR = 10 \cdot \log_{10} \frac{\log_{10} L_{max}}{MSE}, \quad (2.16)$$

where MSE (mean-square error) is computed as

$$MSE = \frac{1}{N} \sum_{i=1}^N [\log_{10}(\hat{L}_t(i)) - \log_{10}(\hat{L}_r(i))]^2, \quad (2.17)$$

where

$$\hat{L}_t(i) = \max(L_t(i), L_{min}), \quad \hat{L}_r(i) = \max(L_r(i), L_{min}). \quad (2.18)$$

Here, $L_t(i)$ and $L_r(i)$ refer to the luminance of pixel i in the test image and reference image, respectively. In addition, L_{max} is the selected maximum luminance, L_{min} is

the minimum luminance considered, and N is the total number of pixels in the image. Typically, L_{max} chosen to be 10000 as most HDR displays will not exceed this value.

2.4.2 Display-referred metrics

The display-referred metrics assumed that the values in the image take into consideration the absolute luminance emitted from an HDR display when displaying the image. In these metrics, that distortions in the image are assumed to be less visible in the darker image regions, which mimic the characteristic of HVS as discussed in Section 2.1. Below are two example of such metrics.

Perceptually-uniform (PU) encoding

PU encoding [58] is proposed specifically to accommodate quality metrics for HDR quality assessment. This method provides a luminance encoding that makes it possible to use the traditional image quality metrics such as PSNR and structural similarity index measure (SSIM) [59] on HDR images. It is formulated based on the contrast sensitivity function which predicts the HVS's sensitivity to light at different luminance values. After applying PU encoding, the absolute luminance values in an HDR image are converted to an approximately perceptually uniform representation. Traditional image quality metrics can then be applied to the converted values to measure the quality.

Visual difference predictor(VDP)

The HDR-VDP [54] metric was the first metric designed for HDR content. It is extended from Daly's VDP model [60] that utilizes a light-adaptive contrast sensitivity function (CSF). The CSF is important as the range of light adaptation can vary substantially in an HDR image. The HDR-VDP is then extended to HDR-VDP-2 [55].

Specifically, HDR-VDP-2 adopts the optical and retinal properties models based on intra-ocular light scattering, the spectral sensitivity of photoreceptors, luminance masking, and achromatic response [4]. Then, it is followed by multi-scale decomposition with steerable pyramids. By using this metric, it is possible to calculate the per-pixel probability maps of visibility and the predicted quality metric [53]. HDR-VDP-2 also takes into consideration the angular resolution when measuring quality.

2.5 Tone mapping operator

As discussed in previous sections, HDR is gaining popularity nowadays but most display devices available in the market are still operating in LDR. The difference between the wide range of luminance that can be captured and the small range of luminance that can be rendered by existing devices makes accurate rendering of the captured scene a problem. This is where tone mapping operators (TMO) comes in. Specifically, TMO transforms the captured HDR to LDR correctly, by considering characteristics of the human visual system and the brightness that can be produced by the device.

There are various TMO techniques. Despite the differences, TMOs can be broadly classified into four groups [43] as follows:

- **Global operator:** The simplest functions that reduce an image's dynamic range by handling each pixel independently;
- **Local operator:** The mapping of a pixel value depends on its neighbours;
- **Frequency operator:** The lower and higher frequencies of the images are processed differently to preserve fine details, and;
- **Gradient operator:** Illuminance and reflectance of an image are taken into consideration in the mapping process.

Global operator

Global operators are computationally efficient. They are generally faster than any other classes of operators due to its simplicity. Real time applications that require speed should consider these operators over others. However, if the dynamic range of an image is extremely high, the global TMO technique may not preserve visibility as good as operators from the other three groups. Examples of global TMOs are Drago Logarithmic Mapping [61] (see Figure 2.5.1b) and Histogram Adjustment [62].

Local operator

Local operators improve the quality of the tone mapped image by attempting to reproduce both the local and the global contrast. This is achieved by having the operator to consider the local statistics of the neighbourhood around the pixels of the image being tone mapped. However, the neighbourhood needs to be defined carefully or the tone mapped image will be distorted by many artifacts. Examples of local operators include Reinhard et al.'s Photogenic Tone Reproduction (which also includes the global mode) [26] (see Figure 2.5.1c) and Ashikhmin's Spatial Variant Operator [63] (see Figure 2.5.1d).

Frequency operator

Frequency operators have the aim of preserving the edges of the tone mapped image. In this case, dynamic range compression is applied in the frequency domain instead of the spatial domain. Generally, low frequency components (which relate to large general features) are tone mapped while high frequency components (which relate to detailed textures) are processed lightly or unaltered. The main characteristic of such methods is that edges and local contrast are preserved if the complete separation between large features and detailed textures can be achieved. Example of frequency operator is

Durand’s Bilateral Filtering [64] (see Figure 2.5.1e).

Gradient operator

Similar to the frequency operator, gradient operator operates in the frequency domain. By taking into account the characteristic of high and low frequency components, it is possible to partially distinguish illuminance and reflectance by considering the gradients in the image. An example of the gradient operator is Fattal’s Gradient Domain Compression [65] (see Figure 2.5.1f).

2.5.1 Inverse tone mapping operator

Since most currently available contents in circulation are in encoded in the LDR format, it is equally important to consider the option to upscale them for display on HDR-capable display device. The technique to convert an LDR image to produce an HDR representation is called ‘inverse tone mapping’. With an inverse tone mapping operator (ITMO), the dynamic range of an LDR content is expanded. Generally, several issues need to be considered when designing a ITMO. Firstly, the shape of the curve to accomplish the dynamic range expansion needs to be determined. Next, under- and over-exposed areas in the image need to be treated to fill in the missing information. Last but not least, expansion of dynamic range may lead to visible banding artifacts [66].

Usually, an ITMO proceeds by taking the following steps [2]:

1. **Linearization** - to create a linear relationship between real-world luminance values and digital pixels.
2. **Expansion of pixel values** - to increase the dynamic range of the LDR image by compressing low values and expanding high values, while keeping mid value intact.



(a) Original

(b) Drago



(c) Reinhard

(d) Ashikhmin



(e) Durand

(f) Fattal

Figure 2.5.1: Examples of tone-mapped images obtained by using different TMOs. The original image (a) is displayed linearly without tonemapping, which results in regions which are too bright (window area) and too dark region (car shadow). Therefore, TMOs are needed to ensure these region can be properly displayed, as shown in (b)-(f).



Figure 2.5.2: Banding due to unsuitable dynamic range. [13]

3. **Over-/under-exposed reconstruction** - to generate the missing (clipped) content in the over-/under-exposed areas of an LDR image.
4. **Artifacts reduction** - to decrease visible artifacts caused by quantization or image compression.
5. **Colour correction** - to the keep original colours as in the LDR image.

One of the earliest ITMO is proposed by Landis [67], which is a power function for luminance expansion. Another example of ITMO is proposed by Akyüz *et al.*, where the authors conducted experiments to observe the effect of different TMOs before designing their ITMO based on linear scaling.

2.6 Data hiding

Data hiding, also commonly known as information hiding, is a term that refers to the act of inserting data into a cover medium, enabling user to later extract the inserted data for various purposes. Data hiding can make the inserted data imperceptible or in other words concealing the existence of the message [68]. Figure 2.6.1 shows the general idea of imagery data hiding and extraction. Digital data hiding is made possible due to

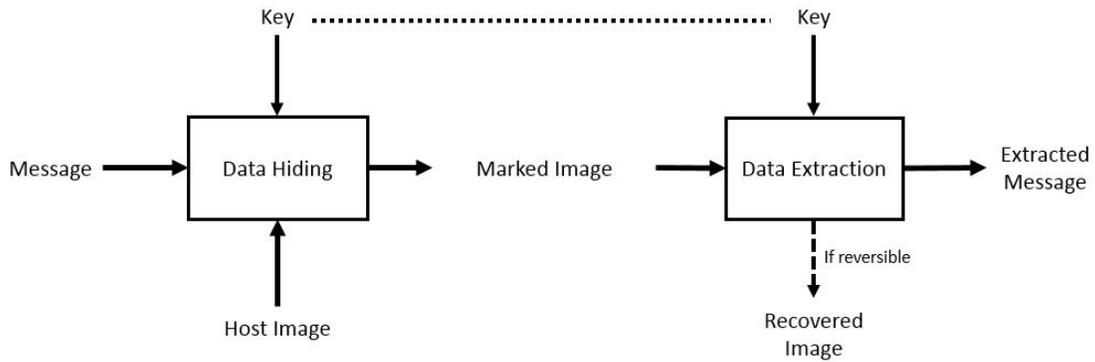


Figure 2.6.1: Data hiding and extraction process

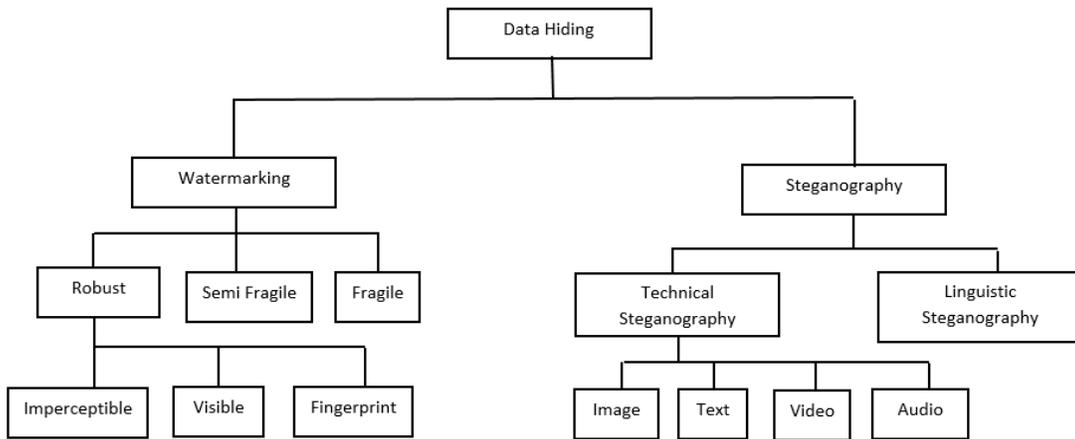


Figure 2.6.2: Different types of data hiding

the nature of digital files. They can be altered to a certain extent without affecting their functionality because human cannot distinguish minor changes in the perceived quality such as changes in intensity value or colour. Typically, data hiding can be divided into a few branches as shown in Figure 2.6.2.

Watermarking

Watermarking, a branch of data hiding, hides information of the owner for ownership verification purpose. The hidden data is called a *watermark*. By design, the inserted watermark should be inseparable and irremovable from the cover medium, even when

its encoding format is changed [68,69].

A well designed watermarking scheme should be able to insert a large amount of information without distorting the quality of the host content. Watermark can be further separated into two main categories, namely, robust and fragile. Robustness refers for the capability of the watermark to survive intentional and unintentional modifications to the image. On one hand, intentional modifications include the removal of the inserted watermark or adding other information without permission to corrupt the inserted watermark. On the other hand, unintentional modifications include processing operations such as compression, scaling, and cropping which are widely utilized.

On the opposite end of the spectrum, as its name implies, fragile watermark is not robust and in fact easily breakable. Fragile watermarking techniques are concerned with integrity verification or in other words, for authentication purposes. Therefore, even the smallest modification of the cover medium can destroy the fragile watermark. This type of watermark is usually inserted to detect modifications made to the host image, and potentially localized the modified regions.

Researchers then consider somewhere between robust and fragile watermarks, and the name semi-fragile watermark is coined. Semi-fragile watermark allows some image processing operations but the embedded watermark can still be utilized to detect unauthorized modification. In other words, it can differentiate normal image processing from malicious content modification [68, 70].

Traditional applications of watermarking include:

1. Content authentication

Data such as signature information can be hidden in content which can be used later to detect and potentially localized tampered regions.

2. Content management

Digital contents can be managed by embedding hyperlink information in them to

facilitate efficient search and retrieval.

3. Fingerprinting

Unique watermark hidden in each copy of a digital content can be used to identify people who illegally redistribute a legal copy. The hidden watermark is similar to a human fingerprint which is unique for each person.

4. Ownership identification

The identity of copyright holder of a digital content can be embedded as a watermark.

With the advancement of HDR imaging, it is important to design new techniques to apply these traditional applications in the newly available HDR images.

Steganography

Another branch of data hiding is called steganography. It can conceal information in a cover medium in such a way that even the existence of the communication is not known to any third party. Steganography can be broadly divided into two types: technical and linguistic [71]. Technical steganography uses technical (physical or chemical) means to conceal the existence of a message, while linguistic steganography uses written natural language to hide data.

Steganography is getting popular in the society and software are being developed for users to transfer secret message between themselves. For example, recently researcher from Columbia University have developed a software called FontCode which can hide data in text through small changes in shapes of letters. FontCode secret messages can transcend from digital to physical and vice versa [72]. Steganography might also be used in crimes. For instance, an engineer exploit steganography on personal images before emailing the images to himself to steal confidential data from the company [73].

Furthermore, extremist groups have developed their own software called 'MuslimCrypts' to spread secret messages in image files [74].

Data hiding in sensitive applications

In both watermarking and steganography, invisibility of embedded data is an important characteristic. Generally after data hiding, the cover digital medium will have some distortion due to the inserted data and cannot be reverted back to original. Permanent distortion has occurred to the cover media even after the extracting the hidden data. In sensitive applications such as medical diagnosis and law enforcement, it is important to the marked medium back to the original after retrieving the hidden data due to legal considerations. For applications such as remote sensing and high-energy particle physical experimental investigation, it is also required that the original cover medium can be recovered due to the high-precision nature [75]. The data hiding techniques that meet this requirement are referred to as *reversible data hiding* techniques. In this thesis, the methods proposed mainly focus on reversibility.

2.6.1 Traditional data hiding

Data hiding can be carried out in two broad domains, namely the spatial domain and frequency domain. In spatial domain, data can be embedded directly by modifying pixels in the image. However, the information hidden might be easily detected by simple statistical analysis and the information might be altered or stolen. To overcome these shortcomings, researchers started to consider the frequency domain. Specifically, the host image will undergo some transformations including the common ones such Discrete Cosine Transform (DCT), Discrete Fourier transform (DFT), and Discrete Wavelet Transform (DWT) in preparation for data insertion. By hiding data using the coefficients, the data is spread across a group of pixels. However, excessively modified

coefficients in the frequency domain will perceptually distort the image.

Data hiding techniques designed for the spatial domain is of low computational complexity and they can survive cropping attack, but they are vulnerable against image modification technique such as salt and pepper noise or compression. On the other hand, data hidden in the frequency domain can usually tolerate some modifications but they are vulnerable against cropping attack. In order to overcome shortcomings in both domains, combining techniques in both domains can have higher robustness and lower complexity [69].

Data Hiding techniques can also be divided into two basic types based on reversibility, i.e., irreversible and reversible. Here, reversibility refers to the ability to recover the original image after data extraction. For the case of irreversible data hiding, high payload can usually be achieved at the cost of not being able to recover the original host image. On the other hand, as discussed before, reversible data hiding (RDH) is crucial for applications where the host image needs to be restored completely after data extraction, for example, medical images, military imaging, as well as evidence used in court. Some examples of reversible data hiding methods are proposed in [75–77]. Between the two extremes, there is another type of data hiding technique called rewritable data hiding. This class of methods preprocess the host image before data hiding, where the preprocessing operations are not invertible in general. Then, the processed image can be used for reversible data hiding. The marked image can be recovered up to the processed image then it can be used for another round of data hiding without causing further distortion. Hence, the name rewritable is coined. Some examples of the existing rewritable data hiding method are proposed in [78, 79].

In addition to the reversibility, embedding capacity is also an important criteria to consider when designing a data hiding algorithm. Embedding capacity, also referred to as payload, is the total number of bits that can be embedded into an image. The

general goal is to maximize payload and robustness while minimizing the distortion introduced due to hiding data. For the rest of this section, the representative data hiding techniques are reviewed.

Least Significant Bit (LSB) Modification

LSB data hiding method involves modification of one bit at each pixel position to embed data. When LSB of a pixel is modified, the change will be bounded by ± 1 , which will not be noticeable by human perception. More bits can be considered, but the image distortion will become noticeable. This is because modification on a bit plane closer to the MSB will cause significant change to the pixel value, causing a significant change to colour intensity of the pixel. An example of inserting 3 bits of data in LSBs is shown in Figure 2.6.3, where the value changes from 179 to 181.



Figure 2.6.3: Example inserting '101' to LSBs.

LSB data hiding method can achieve high insertion capacity since every pixel in the cover image can be the container to host data. However, data hidden in the LSB may be easily detected and extracted by third party if the existence of the message is known. Due to the direct manipulation of every pixel in the cover image, LSB modification method is irreversible and is vulnerable to attacks such as cropping and compression, as well as noise.

Difference expansion

In [76], Tian introduced a reversible data hiding method called difference expansion (DE). This method can achieve high capacity as it utilizes the difference between neigh-

boring pixels, and at the same time achieve reversibility. One bit of data can be hidden into each pair of pixels, therefore DE can achieve a maximum capacity of 0.5 bit per pixel. First the integer average l of a pixel pair and their difference h are calculated as

$$l = \left\lfloor \frac{x + y}{2} \right\rfloor \text{ and} \quad (2.19)$$

$$h = x - y, \quad (2.20)$$

where x and y stands for pixel value and $\lfloor \cdot \rfloor$ denotes the floor function. In order to hide one message bit $b \in \{0, 1\}$, the new difference h' is computed as

$$h' = 2h + b. \quad (2.21)$$

h' is the expanded value of h and subsequently, the new value of the pixel pair (x', y') can be obtained from equation 2.22 and 2.23.

$$x' = l - \left\lfloor \frac{h'}{2} \right\rfloor \quad (2.22)$$

and

$$y' = l + \left\lfloor \frac{(h' + 1)}{2} \right\rfloor. \quad (2.23)$$

However, due to the limited range of 8-bit integer, underflow or overflow problem may occur when computing the new pixels x' and y' . Therefore, some preprocessing steps are needed to be performed on the cover image before hiding data by using DE. The author categorized the pixel pairs based on their expandability or changeability.

The work by Tian [76] lays the foundation for many other reversible data hiding methods. Alattar [80] further improved Tian's method by deriving an enhanced difference expansion transform that is based on a quad, i.e., a set of four pixels, to hide

three bits in every quad. The method is proven to be able to hide the largest payload at the highest signal-to-noise ratio (SNR). Kim *et al.* [81] also proposed a difference expansion method with the simplified and reduced size of location map. Kim's method derived a new expandability which can achieve higher embedding capacity while keeping distortion almost the same as the original method by Tian *et al.*

Another extension of DE proposed by Thodi *et al.* [77], called prediction-error expansion (PEE), has also attracted considerable attention. Unlike in DE where only the relationship between two adjacent pixels is considered, PEE exploits the spatial redundancy in natural image by considering the local correlation of larger neighborhood and therefore can achieve better performance. Instead of the difference operator in difference expansion, PEE usually uses a pixel predictor and data bits are embedded into the prediction errors. PEE has inspired many other works [82–84] for reversible data hiding.

Histogram shifting

Histogram shifting (HS) is another simple yet effective reversible data hiding technique. The histogram of the image pixels is first generated then modified for data hiding. Ni *et al.* [75] are the first group to propose this method. First, a zero point and a peak point are determined from the histogram. Specifically, zero point is a pixel value where its number of occurrences in the image is minimal. On the other hand, a peak point corresponds to the pixel value which has the maximum number of occurrences in the image. As the number of bits that can be embedded into an image (i.e., capacity) depends on the number of pixels in the peak bin $h(M)$, the utilization of the peak point maximize the embedding capacity. With the zero (Z) and peak (M) points identified, the pixels $P(i, j)$ are scanned again in the same order and modified to $P'(i, j)$ according to equation 2.24.

$$P'(i, j) = \begin{cases} P(i, j) - 1, & \text{if } P(i, j) < M; \\ P(i, j) - b, & \text{if } P(i, j) = M; \\ P(i, j), & \text{otherwise.} \end{cases} \quad (2.24)$$

At the extraction stage, the embedded data can be extracted and the image can be restored to the original version by reading the marked pixel values with reference to M and Z . Specifically, when $P'(i, j) < M - 1$, there is no hidden data in this pixel and it can be restored with the expression $P'(i, j) + 1$. When $P'(i, j) \in \{M - 1, M\}$, the pixel contains data and its original value is M . The embedded bit can be obtained as $b = M - P'(i, j)$. When $P'(i, j) > M$, the pixel is unchanged during the data embedding process and no further action is needed.

Similar to difference expansion, PEE has also been widely used in histogram shifting. With PEE, the histogram of prediction errors is utilized instead, and high payload can be achieved through modification of the prediction error histogram. The basic steps of HS-PEE are [25]:

1. Pixel prediction and histogram generation. Prediction error e is computed similar to the typical PEE method discussed above. Then prediction error histogram is generated. Typically, the prediction error histogram has a Laplacian-like distribution centered at 0.
2. Histogram modification. Data is embedded by shifting the prediction error histogram as expressed in Eq. (2.25),

$$e' = \begin{cases} e_i - T, & \text{if } e_i \in [-\infty, -T), \\ 2e_i + m, & \text{if } e_i \in [-T, T), \\ e_i + T, & \text{if } e_i \in [+T, \infty) \end{cases} \quad (2.25)$$

where T is a capacity-dependent integer-valued parameter.

Based on the two steps of HS-PEE, there are two trends of improving the HS method. One is to improve the prediction method to generate a more sharply distributed prediction-error histogram, another one is to design an optimal bins-selection mechanism for HS-PEE. In [85–89], improvements on prediction method are proposed. In [90–93], selection of histogram bins is improved.

The next expansion of HS is through integer wavelet transformation (IWT). The idea of data hiding in discrete integer wavelet transform domain is proposed by Xuan et al. [14]. In [14], Xuan *et al.* proposed a high capacity reversible data hiding method based on integer wavelet transform (IWT). The main idea of this method is to embed data into IWT coefficients of middle and high frequency subbands. Specifically, selected middle bit-planes of IWT coefficients are losslessly compressed with arithmetic coding to make space for data embedding. The purpose of utilizing only the middle and high frequency subbands is to enhance the visual quality of the image.

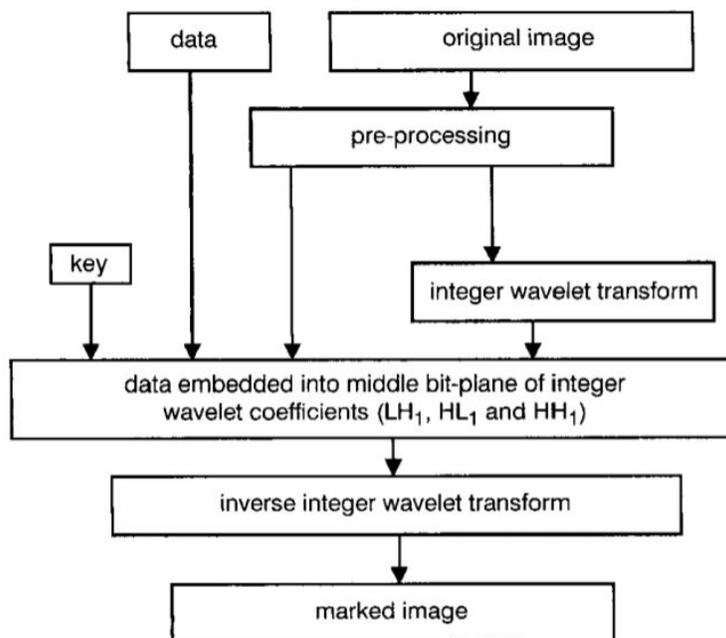


Figure 2.6.4: Distortionless data hiding in transform domain [14].

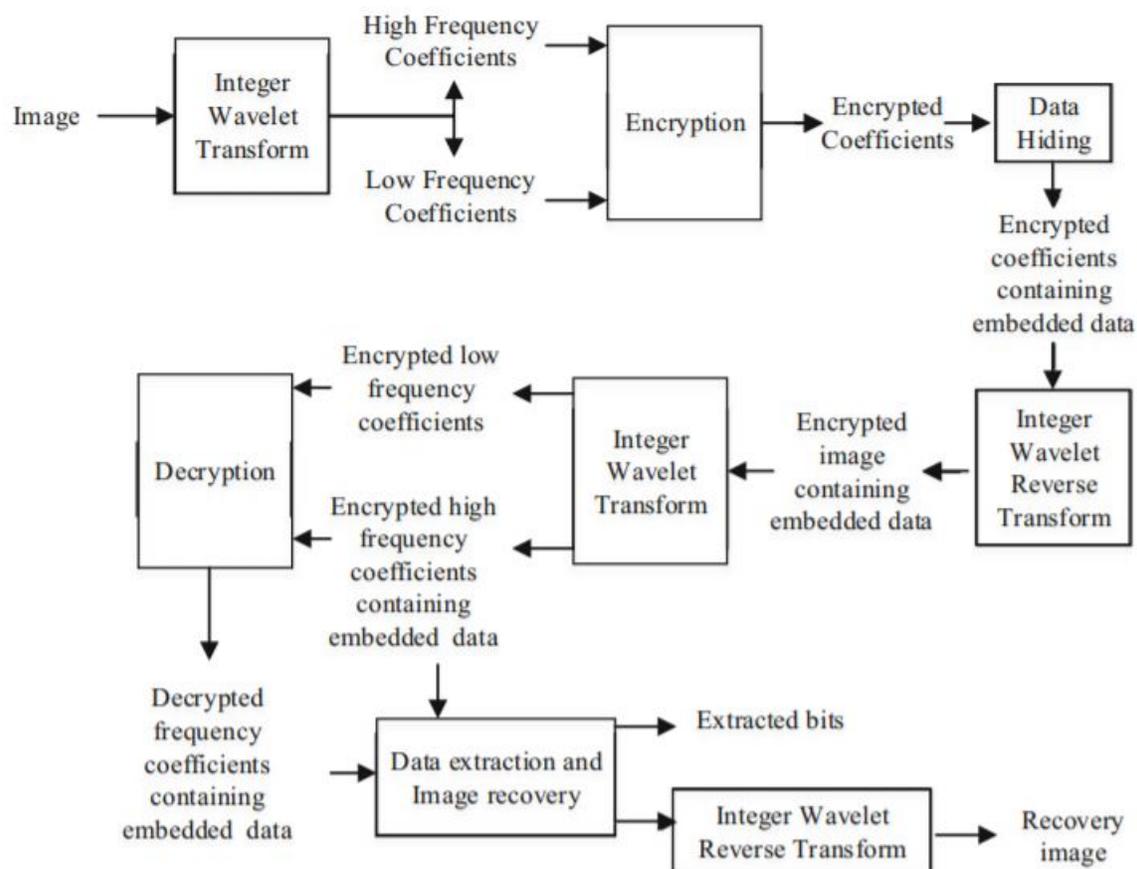


Figure 2.6.5: IWT-HS method for encrypted images [15].

In [94], Xuan et al. apply histogram shifting method on the histogram of high frequency subbands (wavelet histograms). By applying HS in the transform domain, higher payload can be achieved at the same PSNR when compared to traditional HS method. In [15], Xiong et al. proposed a IWT-HS method for encrypted images, as shown in Figure 2.6.5.

2.6.2 HDR data hiding

Traditional data hiding methods cannot be applied directly on HDR image. This is because HDR pixel value is stored in floating point format and hence the nature of floating point representation in computer system (i.e., binary representation) needs to

be considered. Due to this fundamental difference, a few file formats are designed specifically for HDR image, including Radiance RGBE, OpenEXR, LogLUV and JPEG-XT, which are reviewed in Section 2.2.3. Similar to the conventional data hiding methods, it is important to know the characteristic of each file format because it will affect the design of the data hiding technique.

Furthermore, as discussed in Section 2.5, due to the increase in dynamic range, the legacy display devices are no longer able to display HDR images correctly and tone mapping operator (TMO) is invented to solve this problem. TMO maps the wide luminance range of HDR image into a smaller range to approximate the appearance of the HDR image for rendering on a LDR display device. This implies that there can be multiple ways a HDR image is tone-mapped to the LDR image. Therefore, in addition to the traditional applications of data hiding such as watermarking, authentication, hyper-linking images, data hiding in HDR image can also be deployed to guide the tone-mapping process. In addition, the differences between 2 or more tone-mapped images can be embedded into the HDR image via data hiding technology so that the suitable tone-mapped image can be rendered depending on the characteristics of the LDR display device.

The following subsections review the representative data hiding techniques for HDR image.

Histogram shifting-based reversible data hiding for floating point

In [16], Fujiyoshi et al. proposed a simple but efficient histogram shifting-based reversible data hiding (HS-RDH) method for HDR image. The main idea is to treat each floating point pixel value in the HDR image as an integer (i.e., decoding the bit sequence of a floating-point value by following the semantic of an integer) before applying the proposed method. Specifically, histogram of the resulting 16-bit integers are

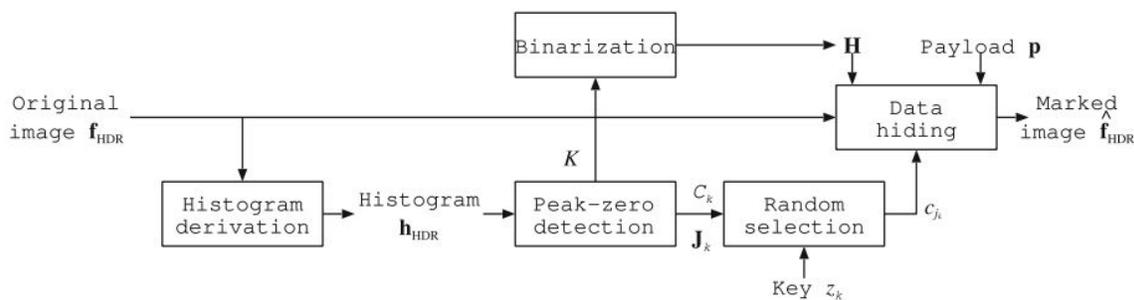


Figure 2.6.6: Block diagram of Fujiyoshi *et al.*'s RDH method [16].

found to be sparse (i.e., many empty bins in the histogram), which is ideal to be used as zero points in HS algorithm. The authors exploited this characteristic and modified HS method to suite HDR data hiding as no preprocessing is needed to place zero point next to the peak point in histogram. In addition, multiple pairs of peak and zero can be used. The proposed HS-RDH method is able to hide data into a HDR image and extract the hidden data without causing distortion. The proposed method is free from memorizing side information by estimating the peak bin in the histogram.

As stated in Section 2.6.1, the original HS algorithm needs to store side information such as the peak point $h(M)$ and zero point $h(Z)$ in order to complete the extraction process at the receiver's side. Fujiyoshi *et al.*'s method eliminates the need of side information in payload extraction, thus allowing more capacity for data hiding. In addition, the method proposed in [16] is blind, which means data can be extracted directly from the processed image without the need to refer to the original image. For a method to be blind, Fujiyoshi *et al.* applied some strategies. Specifically, part of the pixels in a peak bin must remain in the marked image so that the particular bin still remains as the peak bin in the processed image. In other words, the remaining peak after data hiding process must be higher than the next peak and its split. The number of original peak points need to be hidden (i.e., as side information) in the image.

The following steps describe the data hiding process in Fujiyoshi *et al.*'s method:

1. Histogram of HDR image $\mathbf{h}_{\text{HDR}} = h(v_{\text{HDR}})$ is derived.
2. Peak-zero pairs in \mathbf{h}_{HDR} are identified with the condition that zero is followed by peak. The peaks are then arranged, so that the k -th peak is greater than the $(k + 1)$ -th peak, where K is the number of pairs. Let $v_{\text{HDR},0} = \operatorname{argmax} h(v_{\text{HDR}})$.
3. The number of movable pixels for each pair C_k is determined as:

$$C_k = \min \left\{ h(v_{\text{HDR},k}) - h(v_{\text{HDR},(k+1)}), \left\lfloor \frac{h(v_{\text{HDR},k}) - 1}{2} \right\rfloor \right\} \quad (2.26)$$

to ensure that the peaks are still distinguishable after data hiding.

4. With a pseudo random number generated key z_k , C_k from the k -th peak are randomly selected from $h(v_{\text{HDR},k})$. Then, the marked image $\hat{\mathbf{f}}_{\text{HDR}} = \{\hat{f}_{\text{HDR}}(x, y)\}$ is computed as:

$$\hat{f}(x, y) = \begin{cases} f_{\text{HDR}}(x, y) - \alpha_k, & (x, y) = c_{j_k} \text{ and } d(\lambda) = 0 \\ f_{\text{HDR}}(x, y), & \text{otherwise} \end{cases}. \quad (2.27)$$

Here, α_k is the difference between $v_{\text{HDR},k}$ and $v_{\text{HDR},k-1}$. c_{j_k} is j_k -th random element set $\mathbf{J}_k = \{(x, y) | f_{\text{HDR}}(x, y) = v_{\text{HDR},k}\}$, $\lambda = j_k + \sum_{n=0}^{k-1} C_n$, $j_k = 0, 1, \dots, C_k$. $d(\lambda)$ is the λ -th element of \mathbf{d} where \mathbf{d} is the concatenation of \mathbf{H} and \mathbf{p} , where \mathbf{H} is the binary representation of K with length B .

The marked images are evaluated by using quality matrix HDR-VDP-2 [55], in which the result could be a real value up to 100, where 100 represents the highest quality. Results suggest that the algorithm can achieve HDR-VDP-2 of more than 90, with an average payload of 0.0135 bit per pixel per channel. Figure 2.6.7 shows the example tone mapped original and marked images of this method.

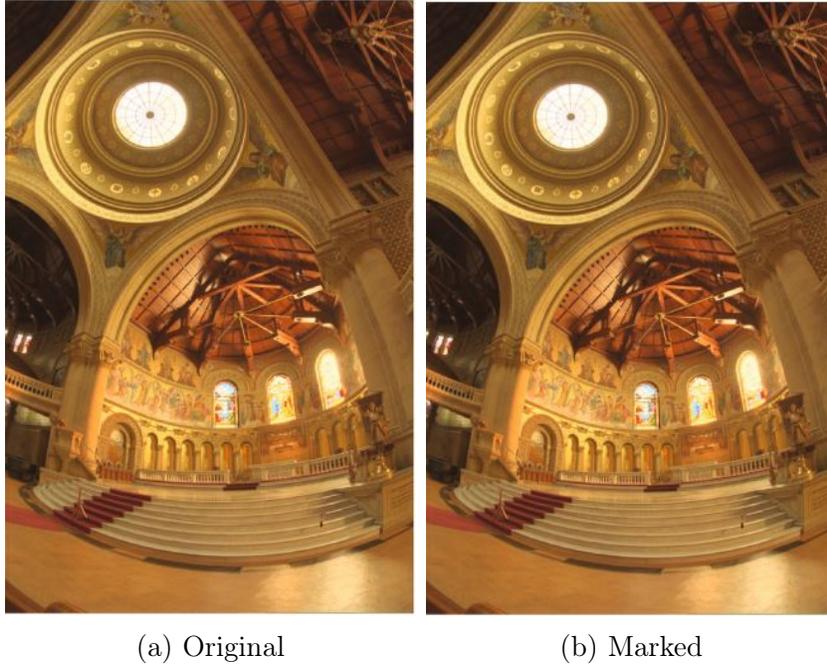


Figure 2.6.7: Example of tone mapped original and marked images from [16].

Data Hiding method for Radiance RGBE file format

Yu et al. [95] first proposed a distortion free data hiding method for Radiance RGBE file format, and it was then improvised by Chang et al. [96]. To achieve distortion free, a group of homogeneous representation is first identified for each pixel (structure of original RGBE pixels is discussed in Section 2.2.3). The key here is to exploit the fact that the common exponent component E used in the radiance RGBE format enables the pixels (R, G, B, E) to be represented in a different manners. Specifically, the group of homogeneous representations can be generated from pixels in two methods:

1. dividing each colour component by two then adding one to the exponent component, i.e., $D(R/2, G/2, B/2, E + 1)$, or
2. multiplying each colour component by two then subtracting one from the exponent component, i.e., $M(R * 2, G * 2, B * 2, E - 1)$.

For example, based on method 1, a RGBE pixel (12, 80, 26, 128) can be converted to (6, 40, 13, 129). On the other hand, based on method 2, they are converted to (24, 160, 52, 127). These values will provide the exact same colour information for display. Division and multiplication operations can be applied to the same pixel recursively as long as the output components remain as integers in the range of [0, 255]. A homogeneity index table (*HIT*) is generated to map message to different values in the homogeneous representation group.

The average capacity of the method is 0.1445 bit per pixel (bpp). As the colour component in the pixels remain unchanged when they are mapped for display, the resulting marked image is free of distortion and the quality of the original image is completely preserved [97].

Wang et al. [98] then proposed a segment-based data hiding scheme based on Yu et al.'s method. Specifically, a group of non-overlapping pixels G in a HDR image are considered collectively as a segment. Homogeneous representations of each pixels are simultaneously multiplied for every segment to generate more homogeneous representations. More homogeneous representations allows the algorithm to insert more data, and vice versa. When using the segment size of $G = 1000$, the average embedding rate is 5% higher than that of Yu et al.'s method.

In [99], Gao et al. proposed a reversible data hiding method for radiance RGBE file format based on histogram shifting by using the generated two-dimensional prediction-error histogram. In the proposed method, two dimensional prediction error histogram (PEH) is utilized in the prediction procedure to organize the prediction errors and maximize the embedding capacity. The exponent channel E represents the range of luminance intensity which is extracted from the other three colour channels and will introduce significant distortion when it is modified to hide data. Due to this characteristic, the E channel is consulted to provide reference to the proposed method.

Furthermore, prediction-error images of different colour channels R ,G and B are very similar among each other, which means the channels have high correlation. This correlation among channels is exploited to further explore the complex regions. In their proposed method, PEH of each channel are generated twice to construct the two dimensional PEH. During the first prediction, data is embedded into pixels in the smooth region. After that, first-time prediction errors in the complex region are then used in the other two channels (reference channels) for second time prediction-errors in one channel (current channel). Then, the two-dimensional PEH is generated from the second time prediction error values to further increase the embedding efficiency and maximize the embedding capacity. Finally, an adaptive embedding strategy designed based on the two-dimensional PEH is employed to reduce the embedding distortion.

Data Hiding method for OpenEXR file format

Lin et al. [100] proposed the first data hiding algorithm for the OpenEXR file format. The method utilizes each of three 10-bit mantissa as the embedding venue to insert k bits of message into a pixel group of n pixels using an optimal base to minimize pixel value variation. Aggressive Bit Encoding and Decomposition (ABED) scheme are deployed to embed data for achieving high probability to convey $(k + 1)$ bits per pixel while minimizing distortion. ABED scheme can read k bits from message bits then check the possibility of exploiting one more bit based on the threshold set, to increase the capacity. Then the selected k or $k + 1$ message bits are decomposed into n bits based on the optimal base generated before hiding them into the pixel. Furthermore, the authors introduced a bit inversion embedding strategy to further increase the capacity when the probability of appearance of message bit ‘1’ is greater than 0.5. Output without perceptual difference can be generated by this method. The authors also show that the proposed algorithm can resist steganalysis from the RS [101]

and SPAM steganalyzers [102].

Recently Bai et al. [17] proposed a reversible data embedding method for OpenEXR file format. Specifically, different combination of bits are extracted from the 16-bit OpenEXR pixel value (in binary representation) to form five 8-bit images (see Figure 2.6.8). The newly formed artificial images are treated as different embedding carriers and subsequently PEE-based data embedding method is applied to these artificial images to embed data. Depending on the complexity of the area, different predictor is applied for PEE to maintain the quality of the image. Furthermore, the embedding strength of PEE can be adjusted according to the exponent value of each pixel to minimize distortion while achieving high embedding capacity.

Table 2.2 summarizes the aforementioned data hiding techniques for HDR image.

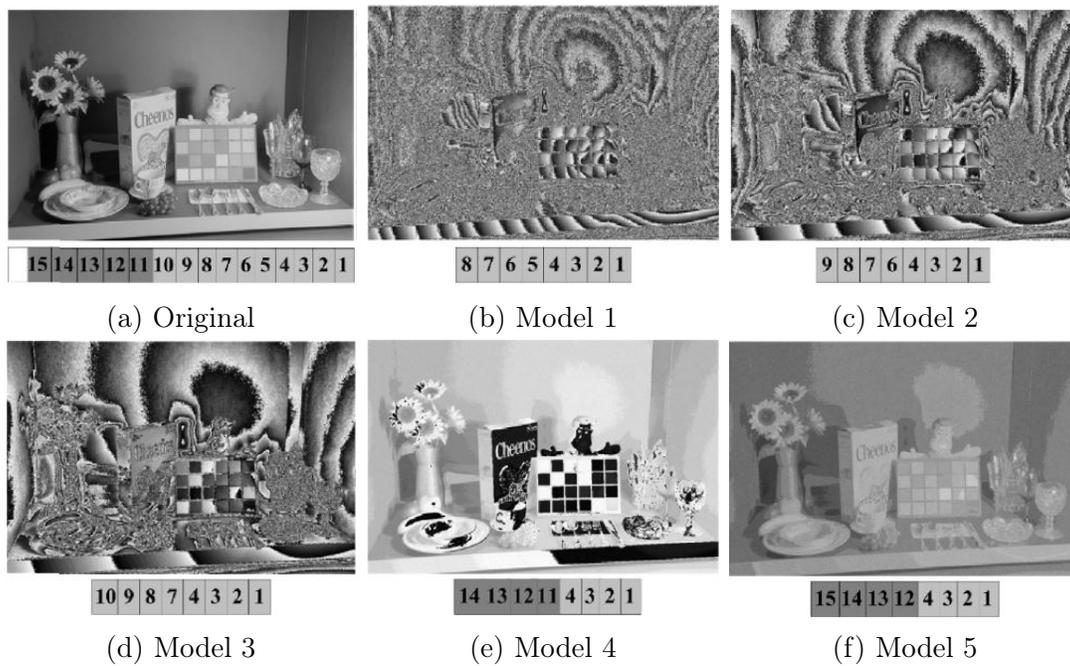


Figure 2.6.8: Model and visualization of the carriers [17].

Table 2.2: Summary of data hiding techniques for HDR image

Reference	HDR file format	Description	Results
Fujiyoshi <i>et al.</i> [16]	Any	Histogram shifting based method which take advantage on the sparse histogram of HDR image	Reversible 0.0135bpp
Yu <i>et al.</i> [95]	RGBE	A distortion-free data hiding algorithm which utilized the characteristic of pixel representation in RGBE.	Irreversible Distortion free 0.1273bpp
Chang <i>et al.</i> [96]	RGBE	Improved version of Yu <i>et al.</i> 's method. A new HIT is designed to increase the efficiency of data embedding.	Irreversible Distortion free 0.1445bpp
Wang <i>et al.</i> [98]	RGBE	Segment-based data hiding scheme based on Yu <i>et al.</i> 's method.	Irreversible Capacity 5% higher than Yu <i>et al.</i> 's
Gao <i>et al.</i> [99]	RGBE	Histogram shifting by using generated two-dimensional prediction-error histogram	Reversible 1.22bpp
Lin <i>et al.</i> [100]	OpenEXR	Utilizes each of three 10-bit mantissa as the embedding venue to insert data bits of message using an optimal base to minimize pixel variation.	Irreversible 2.43–20.00bpp
Bai <i>et al.</i> [17]	OpenEXR	Prediction error expansion-based data hiding algorithm on artificial images generated from bits of the image.	Reversible 0.19–1.06bpp

2.7 Image masking

Perceptual masking aims to mask the semantic of digital content, and it has been heavily relied on for secure communication as well as privacy preservation. As the name implies, perceptual masking aims to distort the perceptual semantic of a content and the output is usually randomized hence resembling noise. Since contents are communicated, shared, and stored online these days, digital content encryption, particularly image, audio and video, have received much attention from the research community [103]. With the emergence of HDR images, there is an increasing need to have proper techniques for managing HDR images and preserving privacy. In that regard, some perceptual encryption techniques are proposed to mask an HDR image. For example, Lin et al. [18] proposed an encryption technique for the RGBE image by using elementary cellular automata (ECA) function. An example of the original, encrypted and decrypted images by Lin et al.'s method are shown in Figure 2.7.1. Yan et al. [104] proposed a format-compliant technique to encrypt the LogLUV TIFF image format.

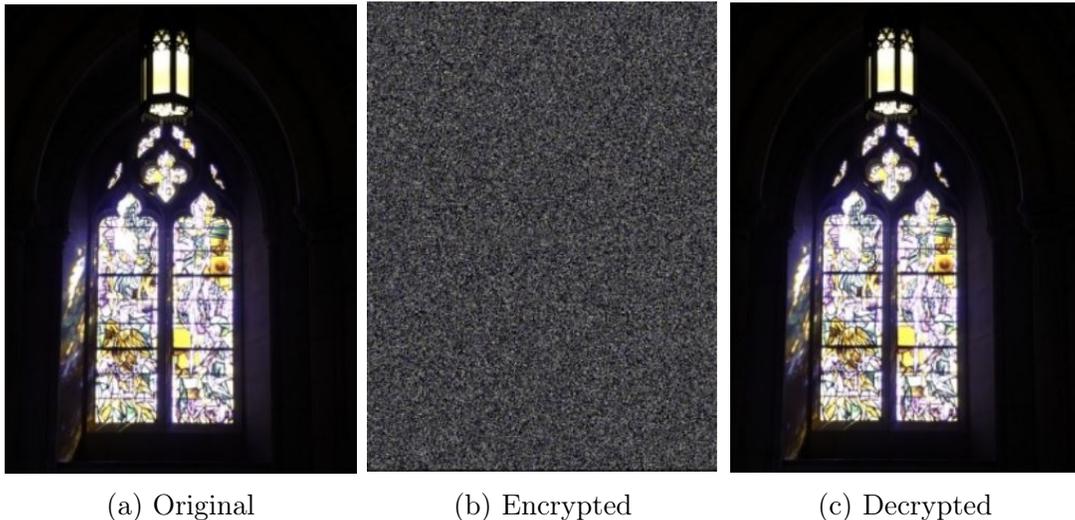


Figure 2.7.1: Example encrypted and decrypted images [18].

2.7.1 Joint data hiding and image masking

Both perceptual masking and data hiding can be jointly utilized for various purposes [105]. To securely share an image, the image can be encrypted by the owner before transmission. In the transmission process, a network administrator may need to add authentication data to the encrypted image without knowing the original image content. For example, in the hospital setting, a nurse may need to extract patients' identification information from the perceptually masked (or encrypted) CT scan images. To protect the privacy of patients, medical images are masked, but for management purposes, patients' information is hidden so that it can be extracted without decrypting the images. Similarly, in the cloud storage, user's contents are encrypted while information about the contents is embedded for administrative purpose. Therefore, the capability to hide data into a perceptually masked HDR image is desired. Some conventional joint techniques include [106–108], but none has been proposed for HDR images.

In [106], the image pixels are encrypted with stream cipher using an encryption key, then additional data can be embedded into the image by modifying a small proportion of encrypted data which are divided into blocks. Each block is further divided into two sets, namely S_0 and S_1 , and the block can be then be utilized to hide one bit by flipping the LSB in S_0 or S_1 depending on the bit to be hidden. For the recovery process, the hidden data can only be retrieved after the image has been decrypted partially with the encryption key, as the data extraction and image full recovery depend on the spatial correlation of the original image. An disadvantage of this method is that there is no way to extract the data hidden before decryption. Therefore, in [107], Zhang improves the method to make it separable, where data can be extracted without revealing the original image content.

In [108], to achieve separability, a specific stream encryption algorithm that can preserve some correlation between the neighboring pixels is proposed. In the proposed

framework, the stream encryption algorithm is followed by a permutation step to the image blocks to completely scramble the image to ensure higher security while preserving the correlation among pixels in the image blocks. After the encryption stage, any previously proposed DHS (difference histogram shifting) and PEHS (prediction-error histogram shifting) based RDH schemes can be used to hide additional information because the correlation among pixels are kept intact after encryption.

2.8 Scalable coding

HDR image require specific graphic card and display device to be rendered correctly. Although the number of HDR media (such as images, movies and video games) is increasing in the past few years, HDR display device is still quite expensive for consumer use. On the other hand, while manufacturers are expected to lower the production cost of HDR display device, most older digital media are in LDR, which in turns not suitable for HDR display. In other words, the hardware capability of HDR display is not fully utilized. Therefore, it is important to have scalable encoding feature to store HDR content and offer backward compatibility feature for LDR device, and vice versa. Backward compatibility offers consumer a way to enjoy HDR contents, even without the required devices/hardware.

Scalable coding can encode HDR image into a single bitstream and based on the capability of the display device, the decoder can extract image with the desired dynamic range to be displayed. For example, in a smaller device with low dynamic range, the decoder can decode the just sufficient information from the HDR content to render the image at the dynamic range that is compatible with the device. When using a better display monitor, the same scene, but represented by using more bits, can be decoded from the same bitstream. Two major factors to be considered when designing scalable

coding are: 1. tone mapping, and 2. colour gamut mapping, which are represented differently in LDR and HDR.

Ward [19] proposed one of the earliest method for backward compatibility of HDR images. The method separates HDR image into two layers, namely, one layer of the tone mapped LDR image, and another enhancement layer with compressed metadata required to reconstruct the image in HDR. Here, the metadata includes luminance ratio of HDR and LDR image, and also colour saturation parameters. During decoding, it is up to the decoder to choose whether to decode the image to HDR or LDR. Figure 2.8.1 and 2.8.2 shows the type pipeline of the encoding and decoding processes, respectively.

Ward's method inspired many more researches on scalable coding on HDR media [20, 109–112]. Mantiuk *et al.* [109] proposed the first backward compatible HDR video compression method, which can ensure the backward compatibility of HDR DVD movies with LDR DVD players. Some representative lossless two layer scalable coding can be found in [20, 111, 112]. For example, Iwahashi *et al.* [111, 112] proposed a lossless scalable coding method for JPEG 2000, which can also be utilized to store HDR image. Specifically, the proposed method reduces the size of the huge enhancement layer by dividing the tone mapping into a reversible logarithmic mapping and its compensation. In [20], Iwahashi *et al.* further improved their method to suite the RGBE and OpenEXR file formats. Figure 2.8.3 and 2.8.4 illustrate the pipeline of the methods in [111] and [20].

2.9 Summary

HDR imaging is becoming a trend nowadays with the release of HDR capable display devices including television and monitor. New file formats, such as Radiance RGBE and OpenEXR, have been proposed to store the new data type of HDR, which is the

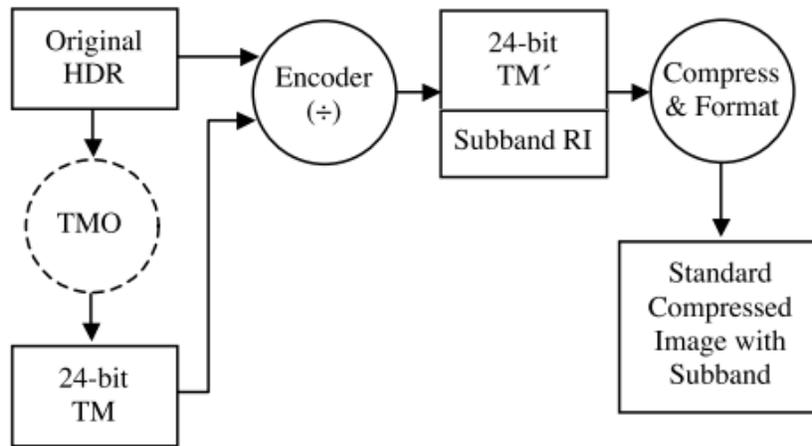


Figure 2.8.1: HDR scalable coding pipeline [19]

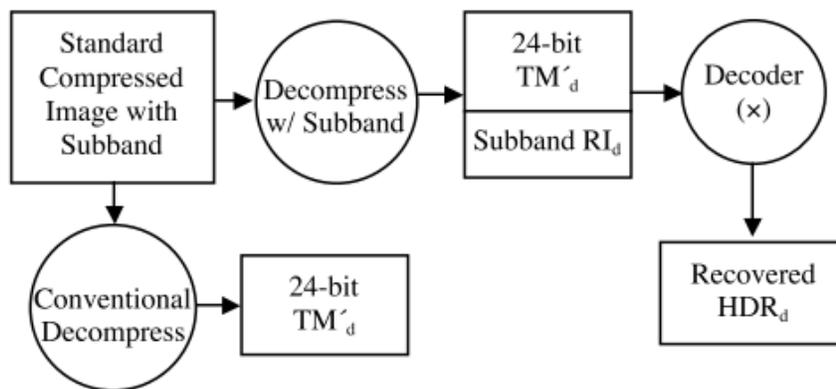


Figure 2.8.2: Decoding paths of the compressed file [19]

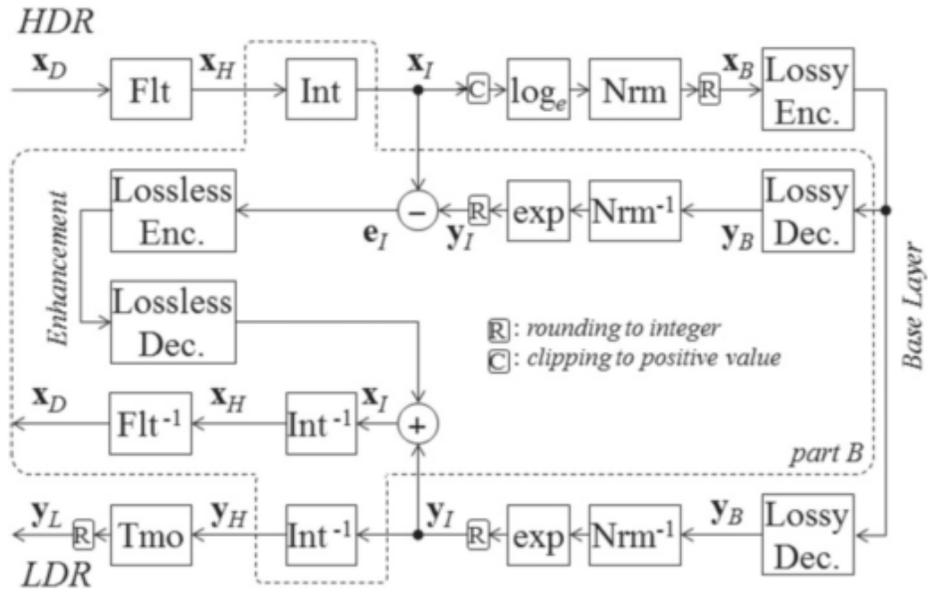


Figure 2.8.3: The baseline method. The HDR image is reconstructed without any loss. Reproduced from [20]

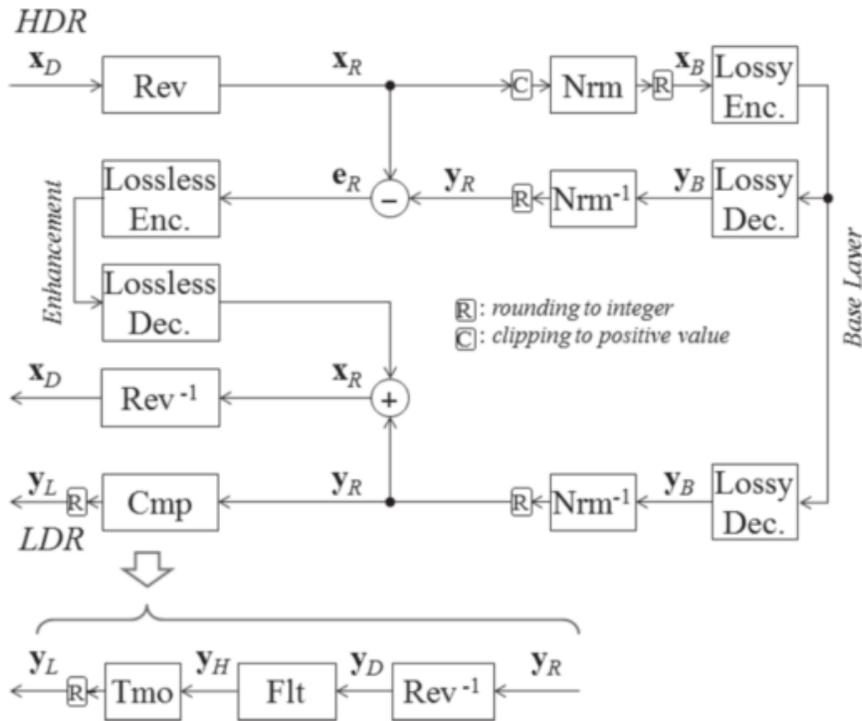


Figure 2.8.4: The proposed type I method. The bit depth of the residual e_R is reduced by the reversible logarithmic mapping Rev . Reproduced from [20]

floating point format. Due to the different structure of HDR image pixels as well as their file formats, traditional methods to protect the images not longer applicable to them. Several new techniques have been proposed recently specifically for HDR images. Furthermore, HDR image cannot be correctly displayed in traditional devices and vice versa. This situation leads to the invention of backward-compatible coding methods for HDR image.

Chapter 3

Reversible Data Hiding in OpenEXR image

This chapter presents two reversible data hiding methods, which are designed based on the the OpenEXR HDR file format. Specifically, the binary representation of floating point value in the OpenEXR file format is manipulated to hide data in both methods. In the first method (HS-RDH), the pixels are divided into groups based on the exponent values and for each group, a histogram is formed for data hiding purpose. In the second method (P-RDH), the pixels are categorized by using a predictor and also based on their natural characteristic before data insertion. Experiments are carried out to verify the performances of both methods and the results show that both method can give high capacity while maintaining perceptual quality. In addition, the images can also be restored to their original states after data extraction.

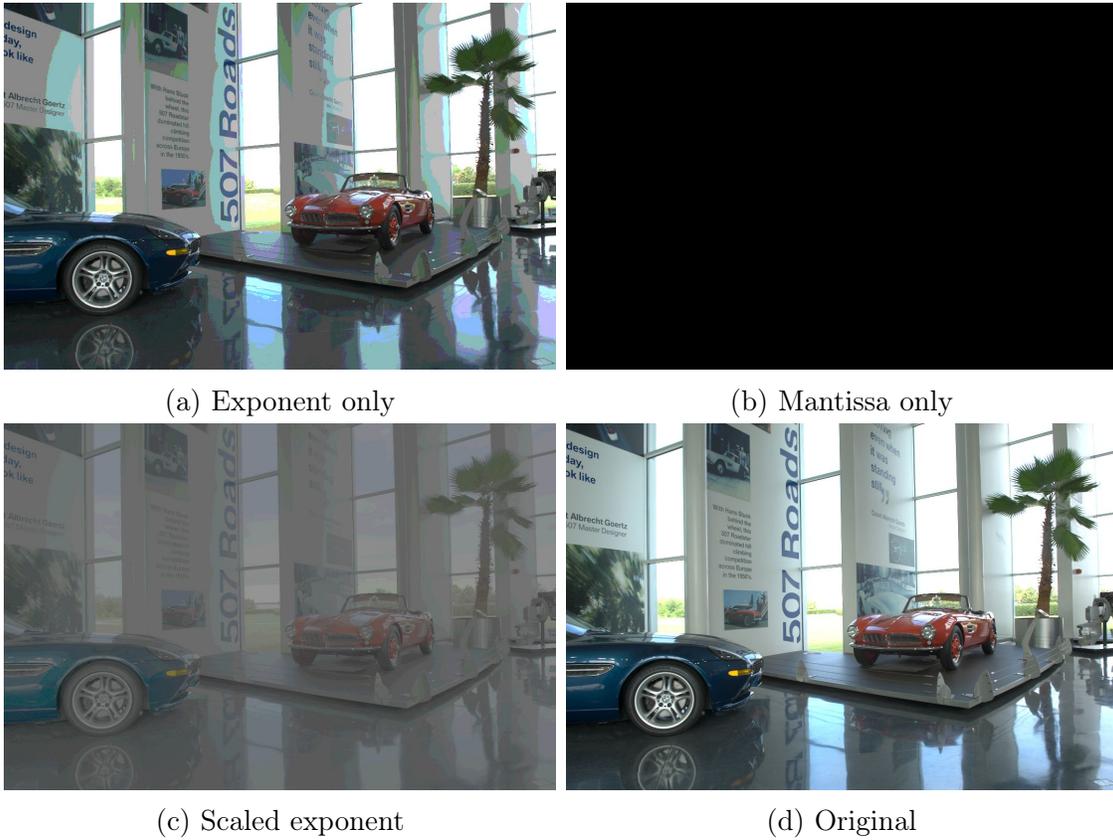


Figure 3.1.1: Example images of different displaying methods.

3.1 Histogram Shifting based Reversible Data Hiding

Based on the layout of half precision format (as discussed in Section 2.2.3), the exponent component of an OpenEXR image captures the coarse value of each pixel, while the mantissa component determines the detailed value of each pixel. An example, Figure 3.1.1 shows the images generated by using: (a) exponent only, where all mantissa values are set to zero, (b) mantissa only, where all exponent values are set to zero, and (c) scaled exponent values, i.e., $E / 31$ to display image in the range of $[0, 1]$. For comparison purpose, the original image is shown in Figure 3.1.1(d). It is obvious that the arrangements (a) and (c) resemble the original image, hence confirming that the role of each component in the representation.

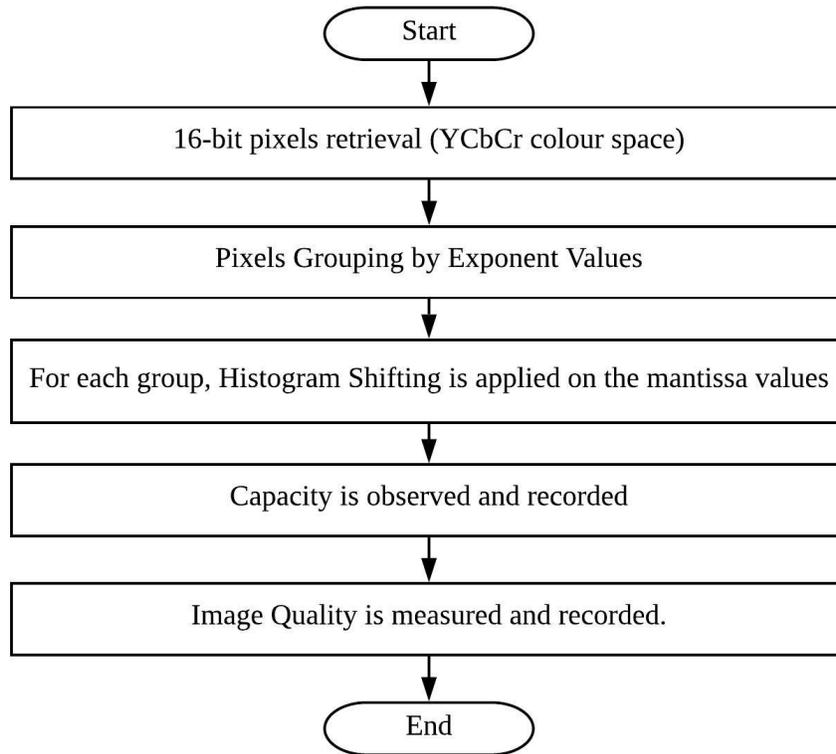


Figure 3.1.2: Proposed method flow.

Based on the aforementioned observations, to suppress distortion caused by hiding data, the exponents are kept unchanged, while the mantissas are manipulated based on the data to be inserted. However, the mantissa values for the all pixels are random (i.e., uniformly distributed) and not predictable. As a result, prediction-based data hiding techniques do not provide good result when applied to the mantissa values. For that, mantissa values are grouped so that each group produces a histogram with some empty bins for the purpose of reversible data hiding.

Figure 3.1.2 shows the flow of processes in the proposed method. First, before processing the image, the colour components are converted from the RGB colour space to the YCbCr colour space using reversible colour transform (RCT) of JPEG2000 as

shown as the equations below [113]:

$$Y = (R + 2G + B)/4, \quad (3.1)$$

$$Cb = B - G, \quad (3.2)$$

$$Cr = R - G. \quad (3.3)$$

Subsequently, the pixels are grouped according to their exponent value. Within each group, the 10-bit mantissa of the pixels then form a histogram. Here, each mantissa is treated as a positive integer, where $M_I \in \{0, 1, 2, \dots, 2^{10} - 1\}$. In essence, there can be up to $2^5 = 32$ independent histograms and the number of elements (pixels) in each histogram can vary from 0 to $D_M \times D_N$, where D_M and D_N refer to the height and weight of the HDR image, respectively. Histogram shifting is applied to hide data.

It should be noted that, among the groups, some may have many pixels and some will have less. In particular, the groups with many pixels suffer from having uniform mantissa histograms and not having any empty bins which are crucial for data hiding using histogram shifting. As an illustration, Table 3.1 the number of pixels for each exponent value is recorded for the image AhwahneeGreatLounge. As expected, the number of pixels in each exponent (hence the corresponding histogram) varies significantly. Therefore, to hide data, the following conditions are set on the group of pixels: (a) mantissa histogram with at least one empty bin, and; (b) at least 10 pixels in the group.

To embed data in the mantissa histogram, the bin next to the peak bin (P) is vacated (i.e., made empty). Specifically, this can be achieved by increasing all mantissa value x where $x > P$. In case the addition operation results in overflow, the opposite can be performed, i.e., decreasing all mantissa value x where $x < P$. Here, the direction of change needs to be communicated to the receiver. Without loss of generality, the

Table 3.1: Number of pixels for each exponent values in *AhwahneeGreatLounge*

Exponent value	Number of pixels
0	138529
1	35083
2	37725
3	43214
4	56172
5	80913
6	139169
7	217674
8	302680
9	352310
10	343739
11	259335
12	169192
13	68944
14	19145
15	8416
16	6090
17	3932
18	4082
19	2029
20	510
21	363
22	546

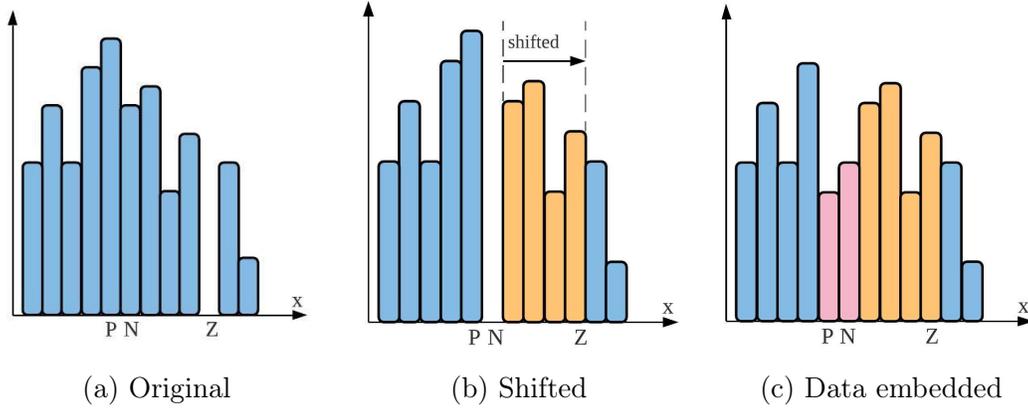


Figure 3.1.3: Illustration of histogram shifting in mantissa component (right shifting).

case of addition is considered. Eq. (3.4) captures the changes made to the pixel value x for hiding data into the mantissa histogram after P is identified and Z located to the right of P and Figure 3.1.3 shows the illustration of the equation.

$$x'(i, j) = \begin{cases} x(i, j), & \text{if } x(i, j) < P \text{ and } x(i, j) > Z; \\ x(i, j) + 1, & \text{if } P < x(i, j) < Z; \\ x(i, j) + 1, & \text{if } x(i, j) == P \text{ and } b = 0; \\ x(i, j), & \text{if } x(i, j) == P \text{ and } b = 1. \end{cases} \quad (3.4)$$

The location of P and Z are then recorded.

In the extraction stage, with the aid of the recorded P and Z , the embedded data can be extracted and histogram of the mantissa component can be restored to the original state. Specifically, when a marked image pixel $x'(i, j) < N - 1$, there is no hidden data in it and it can be kept unchanged. When $x'(i, j) \in \{N, Z\}$, the pixel is shifted and its original value can be restored with $x'(i, j) - 1$. When $x'(i, j) = N$, the embedded bit is '0' and can be restored with $x'(i, j) - 1$. When $x'(i, j) = P$, the embedded bit is '1' and can be kept unchanged. The restoration process is detailed in

Eq. (3.5).

$$x(i, j) = \begin{cases} x'(i, j), & \text{if } x'(i, j) < P \text{ and } x'(i, j) > Z; \\ x(i, j) - 1, & \text{if } P + 1 < x'(i, j) \leq Z; \\ x(i, j) - 1, & \text{if } x'(i, j) == P + 1; \\ x(i, j), & \text{if } x'(i, j) == P. \end{cases} \quad (3.5)$$

3.1.1 Experiments and discussions

The proposed method is implemented in Matlab R2020a. Experiments are carried out with 108 HDR images from the dataset in [21]. Figure 3.1.4 shows six representative images from the dataset. They are chosen because of their diversified characteristic. For instance, image *BalancedRock* and *BrandonSunset(1)* are outdoor natural scenery, *507* and *AhwahneeGreatLounge* are indoor scenery with man made items, and *Cemetery-Tree(2)* and *Frontier* are images with high contrast due to the bright lighting (the sun and the night light) in the scenery. To conduct the experiment, each test image is down-scaled to about $D_M \times D_N = 712 \times 1072$ pixels. It is verified that the inserted data can be extracted and the original image can be obtained (i.e., reversible).

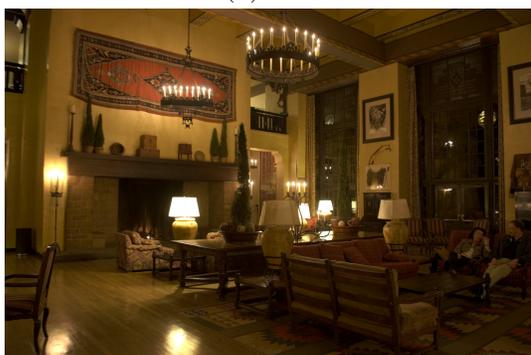
Figure 3.1.6 shows the graph of capacity of each test images in the unit of bit per pixel (bpp). The average achievable capacity is 0.0106bpp, which translate to 1 bit per 100 pixel. Table 3.2 records the capacity of the six test images (Figure 3.1.4) in both total number of bits and bit per pixel (bpp), while Figure 3.1.5 shows the corresponding marked images. Results suggest that the image *BandonSunset(1)* achieves the highest payload (i.e., 0.2793bpp), while the lowest payload is attained by *AhwahneeGreatLounge* (0.0001bpp). The lower capacity is due to the larger differences between the brightest and darkest pixels, causing a larger range of exponent values and therefore more groups (hence histograms) are formed. Since the number of pixels is fixed, more histograms



(a) 507



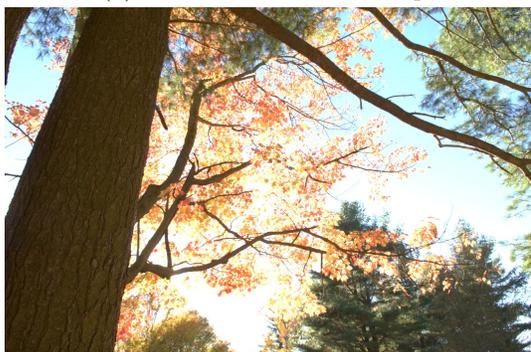
(b) BalancedRock



(c) AhwahneeGreatLounge



(d) BandonSunset(1)



(e) CemeteryTree(2)



(f) Frontier

Figure 3.1.4: Six representative HDR test images from [21]. Each image is tone-mapped here for display purposes.



(a) 507



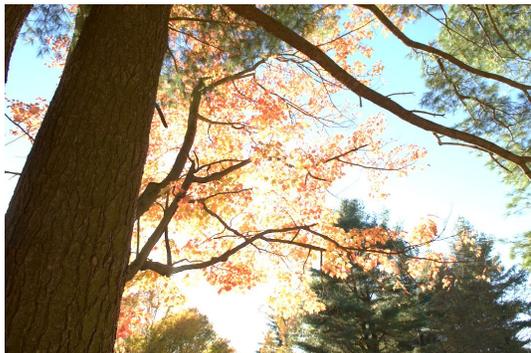
(b) BalancedRock



(c) AhwahneeGreatLounge



(d) BandonSunset(1)



(e) CemeteryTree(2)



(f) Frontier

Figure 3.1.5: Six representative HDR images with data embedded. Each image is tone-mapped here for display purposes.

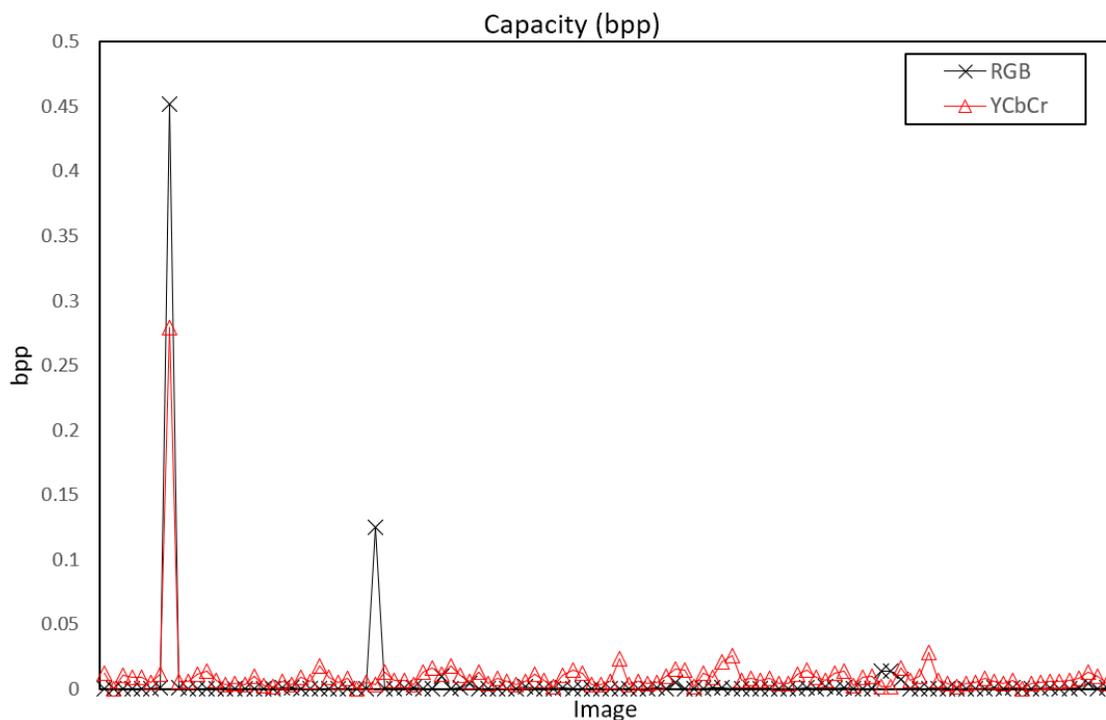


Figure 3.1.6: Capacity of each test images

Table 3.2: Example capacity of six test images.

Filename	YCbCr		RGB	
	Capacity (bits)	Capacity (bpp)	Capacity (bits)	Capacity (bpp)
507	28642	0.0125	816	0.0004
BalancedRock	25824	0.0113	1933	0.0008
AhwahneeGreatLounge	194	0.0001	103	0.0001
BandonSunset(1)	541691	0.2793	875389	0.4514
CemeteryTree(2)	4797	0.0021	188	0.0001
Frontier	7076	0.0031	285243	0.1245

means that each will have lower number of pixels. In other words, the same number of pixels needs to be distributed into more histograms. Histograms with low number of pixels have insignificant peak (i.e., less than 10 pixels) and they are excluded from the data hiding process.

For reference purpose, the payload achieved by using the RGB colour space (i.e., without conversion to YCbCr) for the test images are also included. The average

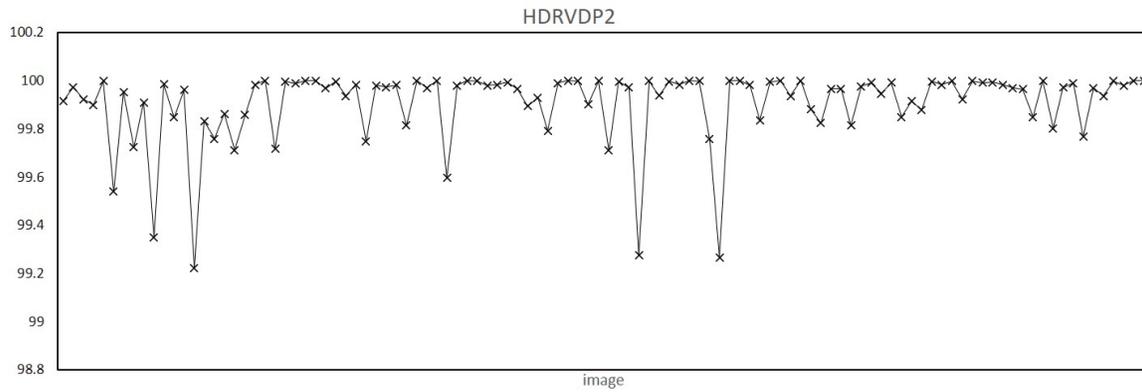
Table 3.3: Quality assessment.

Filename	logPSNR	HDR-VDP-2
507	93.7008	99.8985
BalancedRock	93.0980	99.9464
AhwahneeGreatLounge	91.7360	99.9721
BandonSunset(1)	91.6366	99.7238
CemeteryTree(2)	93.5322	99.8377
Frontier	89.8769	99.9802

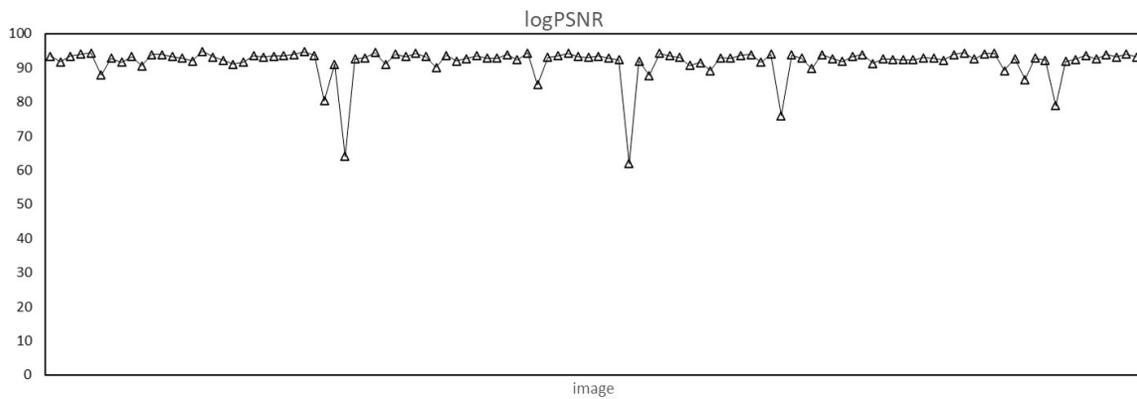
payload achieved without conversion is 0.0062bpp, which is lower in comparison to the payload achieved in the YCbCr colour space. Based on the results, it is concluded that converting RGB to YCbCr leads to higher payload for the proposed method.

Next, the image quality is measured in terms of logPSNR [58] and HDR-VDP-2 [55]. As discussed in Section 2.4.1, the logarithmic function used in logPSNR [58] is an approximation of how human visual system responds to light, which makes this metric more suitable in capturing the quality of HDR image in comparison to the legacy PSNR metric. On the other hand, HDR-VDP-2 [55] incorporates more factors such as human eye reaction, display luminance and viewing distance, thus making it one of the best quality metrics in quantifying image quality. The quality value of HDR-VDP-2 ranges from 0 to 100, with 0 and 100 being the worst and highest qualities, respectively. The results are plotted in Figure 3.1.7 and Table 3.3 explicitly records the values for the six representative images. In general, the logPSNR values for the processed images are high, and the HDR-VDR-2 values are more than 99. These suggest that quality of the processed HDR image (i.e., containing hidden data) is high.

Table 3.4 compares two existing methods with the proposed method. It can be seen that the proposed method can achieve the highest quality but the lowest capacity. This shows that the proposed method can generate high quality marked images, therefore ensuring the imperceptibility of the embedded data. However, low capacity is not practical for application requiring room to host large volume of data. Due to this,



(a) HDR-VDP-2



(b) logPSNR

Figure 3.1.7: Image quality assessment

Table 3.4: Comparison of methods with proposed.

Scheme	Capacity (bpp)	HDR-VDP-2
Fujiyoshi <i>et al.</i>	0.0430	98.55
Bai <i>et al.</i>	0.19~1.06	88~93
Proposed method	0.0062	99.91

another reversible data hiding method for OpenEXR image is proposed in Section 3.2.

3.1.2 Summary and conclusion

A reversible histogram-based data hiding method is put forward for the OpenEXR file format. Specifically, pixels in an HDR image are grouped based on the exponent values, and the mantissa values of each group then form a histogram for the purpose of reversible data hiding. Experimental results suggest that an average payload of 0.0106 bpp is achieved while the distortion caused is small.

3.2 Reversible Data Hiding based on Natural Characteristics of Pixels

Based on the layout of half precision format (see Section 2.2.3), the exponent component of an OpenEXR image captures the coarse value of each pixel, while the mantissa component determines the detailed value of each pixel. Therefore, it is possible to predict the exponents values using a predictor. By utilizing a predictor, a mask is created to identify pixels that are suitable for data hiding, in essence, the pixels that can be recovered using prediction after data extraction. The main steps of the proposed method are shown in Figure 3.2.1.

The first step of the proposed method involves predicting the exponent component of all pixels by using a pixel predictor. The purpose of this step is to identify well

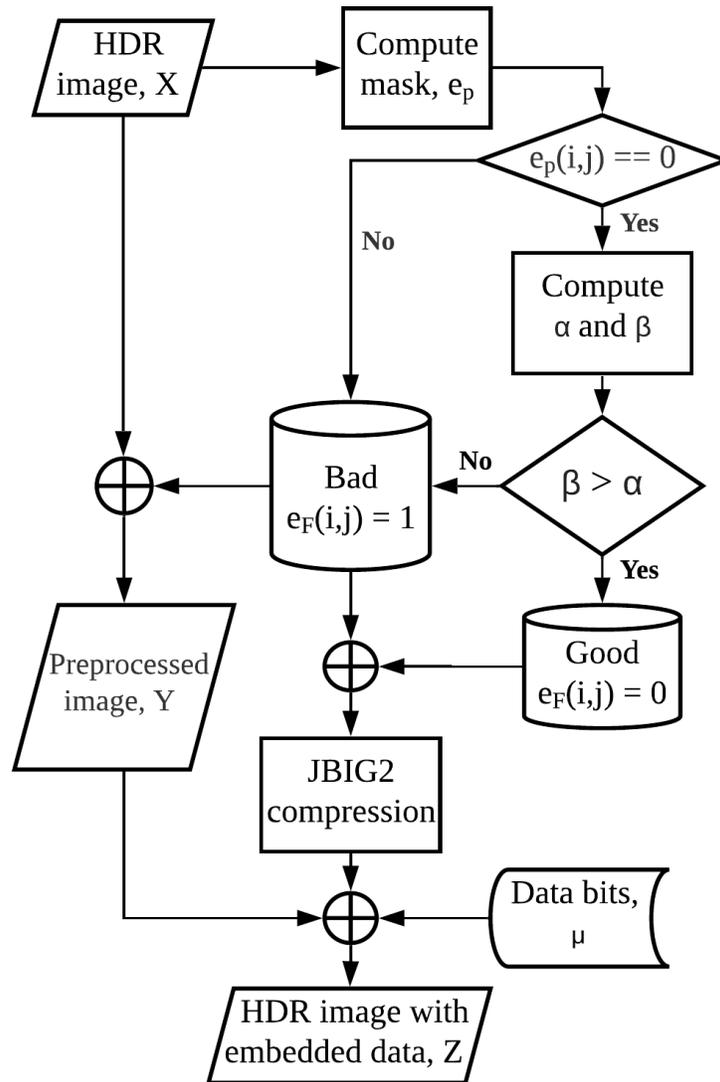


Figure 3.2.1: Flowchart of proposed method

predictable locations, which are used later for data hiding purposes. Note that well predictable pixel locations, to certain extents, implies that the pixel of interest is similar to its neighbors. In this method, the Median Edge Detection (MED) predictor [114] is chosen for simplicity. MED is a simple predictor which relies on the direction of the edges, viz., horizontal, vertical, or no edges at all. The operation of MED is as shown in Eq. (3.6), where X is the value to be predicted, N, W and NW are the top, left and top-left pixels of E , respectively. MED predictor predicts all values following the raster scan order, where pixels at the upper-left border are utilized as the reference pixels. The predicted value of X , denoted by X' , is computed as follows:

$$X' = \begin{cases} \min(N, W), & \text{if } NW \geq \max(N, W); \\ \max(N, W), & \text{if } NW < \min(N, W); \\ N + W - NW, & \text{otherwise.} \end{cases} \quad (3.6)$$

Then, the prediction error is recorded as:

$$e_P(i, j) = \begin{cases} 1 & \text{if } |X'(i, j) - X(i, j)| > \tau; \\ 0 & \text{otherwise.} \end{cases} \quad (3.7)$$

Note that $e_P(i, j) = 0$ indicates a well predicted pixel while $e_P(i, j) = 1$ indicates an ill-predicted pixel. Value of τ will affect the hiding capacity as well as the quality of the output image. Specifically, when τ increases, the capacity increases and the output image quality decreases, and vice versa.

After prediction using the exponent component, another stage of pixel selection is applied on the predictable pixels (i.e., $e_p(i, j) = 0$) by using the natural characteristic of the image. Note that the subsequent steps are only performed on the pixels at position (i, j) where $e_p(i, j) = 0$. Typically, a pixel resembles more to the original pixel when

less modification is apply to it, and vice versa. In essence, the predictability of a pixel using its neighboring pixels decreases if more modification is applied to the pixel, and vice versa. This characteristic of natural image is exploited to hide data. To carry out this step, firstly the MSB of the mantissa component is flipped to achieve the largest possible change to a pixel value, while keeping the sign and exponent components unchanged. With the original pixel X and the flipped version of pixel X^f , the following are computed:

$$\text{Original Sum, OS} = (X - W)^2 + (X - NW)^2 + (X - N)^2 \quad (3.8)$$

$$\text{Flipped Sum, FS} = (X^f - W)^2 + (X^f - NW)^2 + (X^f - N)^2 \quad (3.9)$$

Take note that this operation treats the pixels as a floating point number. Based on the characteristic of natural image, OS should be less than FS. Therefore, in most locations, $OS \leq FS$, but there are some exceptions. Therefore, a location map e_F is generated to record the positions that disobey this natural characteristics. Specifically, $e_F(i, j) = 0$ implies that $OS \leq FS$, while $e_F(i, j) = 1$ implies that $OS > FS$. The location map e_F is then compressed by using JBIG2 to reduce size and treated as data to be inserted later. The compressed location map $\text{JBIG2}(e_F)$ is then combined with the data to be embedded μ to form the augmented data $\mu^a = \text{JBIG2}(e_F) || \mu^a$, where $||$ is the concatenation operation. Therefore, the augment data μ^a is embedded into the image, instead of μ . The image X is preprocessed to produce the image Y so that for each pixel and for each channel $Y_c(i, j)$, the relationship of $OS \leq FS$ holds true.

To embed data, depending on the bit from μ^a to be embedded into the pixel, the MSB of the mantissa is flipped. Specifically, no modification is done to the MSB if ‘0’ is to be embedded, while the MSB is flipped to embed ‘1’. This process is repeated for each pixel and each channel until all pixel locations are considered or when the entire

augmented payload data is embedded. The resulting image is denoted by Z . Note that, prior to MSB manipulated to hide data, the payload data encoded by the preprocessed image Y is a sequence of zeros.

3.2.1 Experiments and discussions

The proposed method is implemented in Matlab R2020a. Experiment is carried out with 108 OpenEXR images from the dataset [21], but each image is down-scaled to 712×1072 pixels (which is similar to the experiment conditions in Section 3.1.1). Unless specified other, $\tau = 1$ (see Eq. (3.6)) is set for experiment purpose.

Figure 3.2.2 shows the graph of capacity vs image name. Here, the raw capacity (i.e., number of bits that can be embedded) is computed as $\kappa_r = |\{e_P(i, j) | e_P(i, j) = 0\}| / |A|$, where $|X|$ refers to the cardinality of the set X and $A = \{A_r, A_g, A_b\}$ refers to the host image. On the other hand, the effective capacity is computed as $\kappa_a = \kappa_r - |M| / |A|$. Here, κ_a indicates the actual number of pixels that are suitable for data hiding, where the effective capacity is the actual space to embed useful information. Results suggest that, on average, $\bar{\kappa}_r = 0.9806$ bit per pixel channel or 2.942 bit per pixel of raw capacity is achieved. The average effective capacity $\bar{\kappa}_a$ is only slightly lower than that of average raw capacity, which is 0.8753 bit per pixel channel or 2.626 bit per pixel. As observed from Figure 3.2.2, the variation of κ_a ranges from 0.7833 to 0.9763 bppc, which is caused by the differences in image texture / characteristics, hence the predictability of the MSB of mantissa. It is noteworthy that the standard deviation of the results is 0.0514.

Next, the image quality is measured in terms of logPSNR [58] and HDR-VDP-2 [55]. The quality value of HDR-VDP-2 ranges from 0 to 100, with 0 and 100 being the worst and highest qualities, respectively. Figure 3.2.3 shows the graph plot of the results. The average logPSNR values for the processed images is high at a value of 51.66

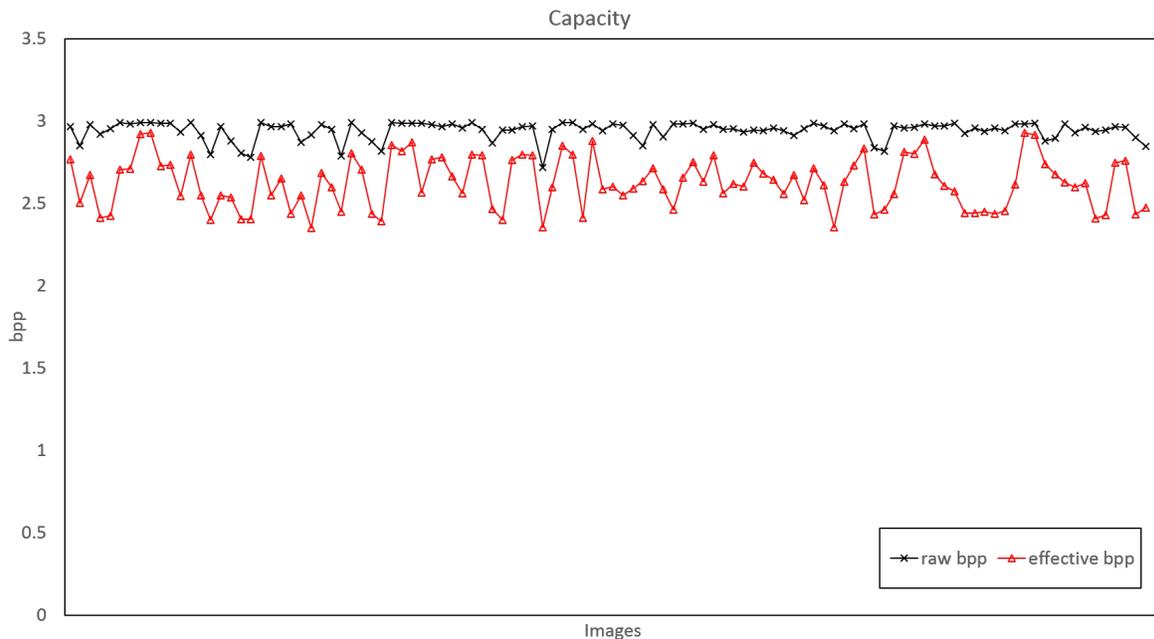
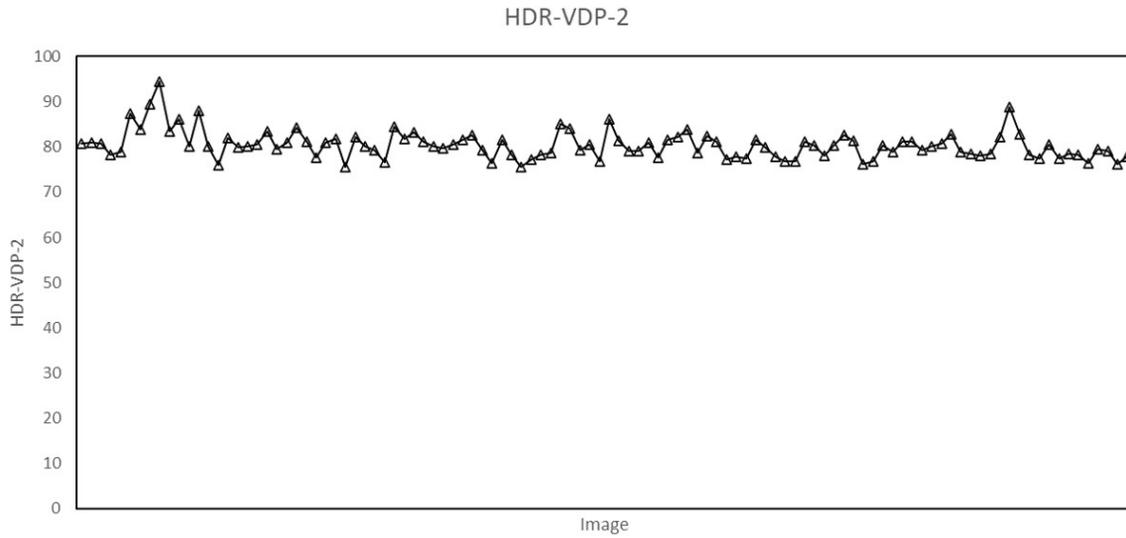


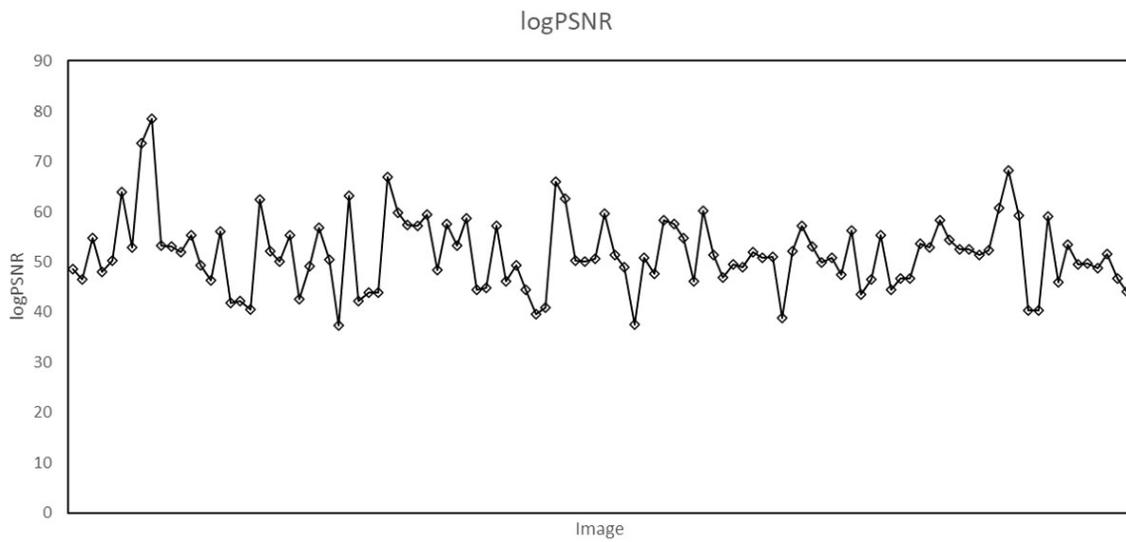
Figure 3.2.2: Number of bits embeddable into each image in the unit of bit per pixel (bpp). The average value of raw capacity is 2.942bpp and the average value of effective capacity is 2.626bpp

($\log\text{PSNR} > 30$) and the average HDR-VDP-2 value is 80.49. These outcomes suggest that quality of the processed HDR image (i.e., containing data) is high. Figure 3.2.4 shows six example HDR images with data embedded and it can be observed that the distortion in each image is low.

Table 3.5 shows the comparison between the proposed method and two conventional reversible data hiding schemes for HDR image. Specifically, Chang et al.'s method [96] achieves an average capacity of 0.1445bpp without distorting the output image. On the other hand, Fujiyoshi et al.'s [16] method achieves an average capacity of 0.0430bpp with an average HDR-VDP-2 value of 98.55 without the need of image-dependent parameter such as location map. For Bai et al.'s method [17], the embedding capacity ranges from 0.19 to 1.06bpp since it can form multiple artificial images to embed data, while the image quality ranges from 88 to 93. On the other hand, the proposed method can achieve an average capacity of 2.626bpp average HDR-VDP-2 value of 80.49. Although



(a) HDR-VDP-2



(b) logPSNR

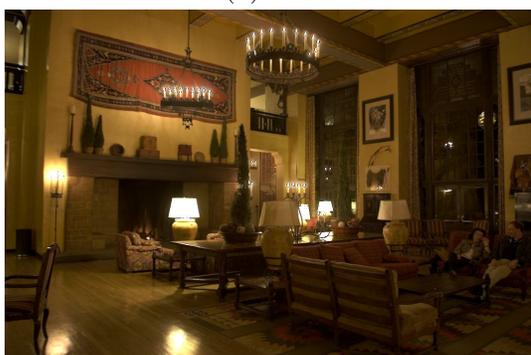
Figure 3.2.3: Image quality assessment



(a) 507



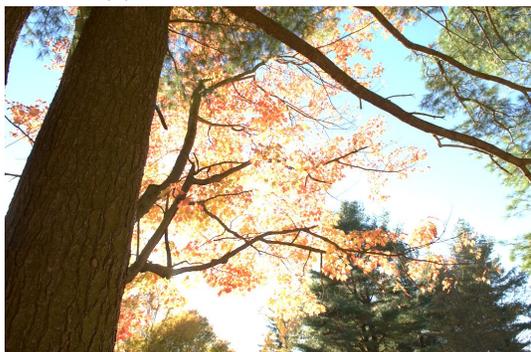
(b) BalancedRock



(c) AhwahneeGreatLounge



(d) BandonSunset(1)



(e) CemeteryTree(2)



(f) Frontier

Figure 3.2.4: Six representative tonemapped HDR images with data embedded to observe the quality.

Table 3.5: Comparison of methods.

Scheme	Capacity (bpp)	HDR-VDP-2
Chang <i>et al.</i>	0.1445	distortionless
Fujiyoshi <i>et al.</i>	0.0430	98.55
Bai <i>et al.</i>	0.19~1.06	88~93
Proposed method	2.626	80.49

the processed image quality of the proposed method is slightly lower, significantly higher capacity can be achieved. In other words, the image quality is traded for higher capacity, but it should be noted that the proposed method is also reversible.

3.2.2 Summary and conclusion

In this method, a reversible data hiding method for OpenEXR file format is put forward. Specifically, the exponent component is predicted and the prediction error is used as a guide to select a group of pixels for data embedding purposes. Then, the characteristics of natural image is analyzed and the pixel locations failing the proposed natural property are recorded for complete reversibility. Finally, data is inserted into the locations at which the exponent component is well predicted. Experimental results suggest that an average payload of 2.626 bpp is achieved while the distortion caused is small.

3.3 Application of the proposed method

In addition to the traditional applications for content management, the proposed method can be deployed to store the differences between 2 tone-mapped images. As discussed above, traditional LDR display device is not capable of rendering HDR image correctly. However, currently most display devices used by typical consumers have standard / low dynamic range (SDR / LDR). Tone mapping operators (TMOs) are designed to

shrink HDR to LDR for proper display of visual content. Some typical TMOs include Reinhard’s TMO [115] and Drago’s adaptive logarithmic TMO [61]. As different devices have different display capability (e.g., brightness, contrast and colour gamut), an HDR image may need different TMO to properly render the image on different display devices, i.e., system dependent. In addition, some system only implement a limited choices of TMO, hence not able to properly display the visual content in the intended manner. The proposed RDH methods can be deployed to store the differences D between 2 tone-mapped images A_1 and A_2 , which are obtained by using TMO_1 and TMO_2 , respectively. An example scenario is illustrated in Figure 3.3.1. In this scenario, one can imagine that TMO_1 is a widely utilized TMO, while TMO_2 is less known or content dependent. This information can aid the decoding process at the user’s end when he/she tries to reconstruct the corresponding LDR image to be viewed on a traditional device without implementing the specific TMO decoder from the content creator (e.g., restricted by license, complexity of operations). As an experiment to realize the application discussed above, the luminance value of each test image (A^Y) is tone mapped by using two TMOs, namely, Reinhard’s TMO [115] (A_R^Y) and Drago’s adaptive logarithmic TMO [61] (A_D^Y). The difference is calculated as $\Delta(A) = A_R^Y - A_D^Y$. Next, 8×8 discrete cosine transform (DCT) is applied to $\Delta(A)$ to obtain the DC coefficients. Finally, the DC coefficients are rounded to integer values. The rounded integer DC coefficients are then converted to signed binary bits (i.e., most significant bit represent sign bit) to be hidden into the image. If the data bits exceed the capacity available, the quantization process can be repeated with the DC coefficients scaled down with a factor F to decrease the number of bits to represent the DC coefficients. It should be noted that the image recovered by using the extracted coefficients is not exactly A_R^Y or A_D^Y , but very similar to them. This is because the AC coefficients are not stored and are lost during the RDH process. Figure 3.3.2 shows the capacity needed for storing

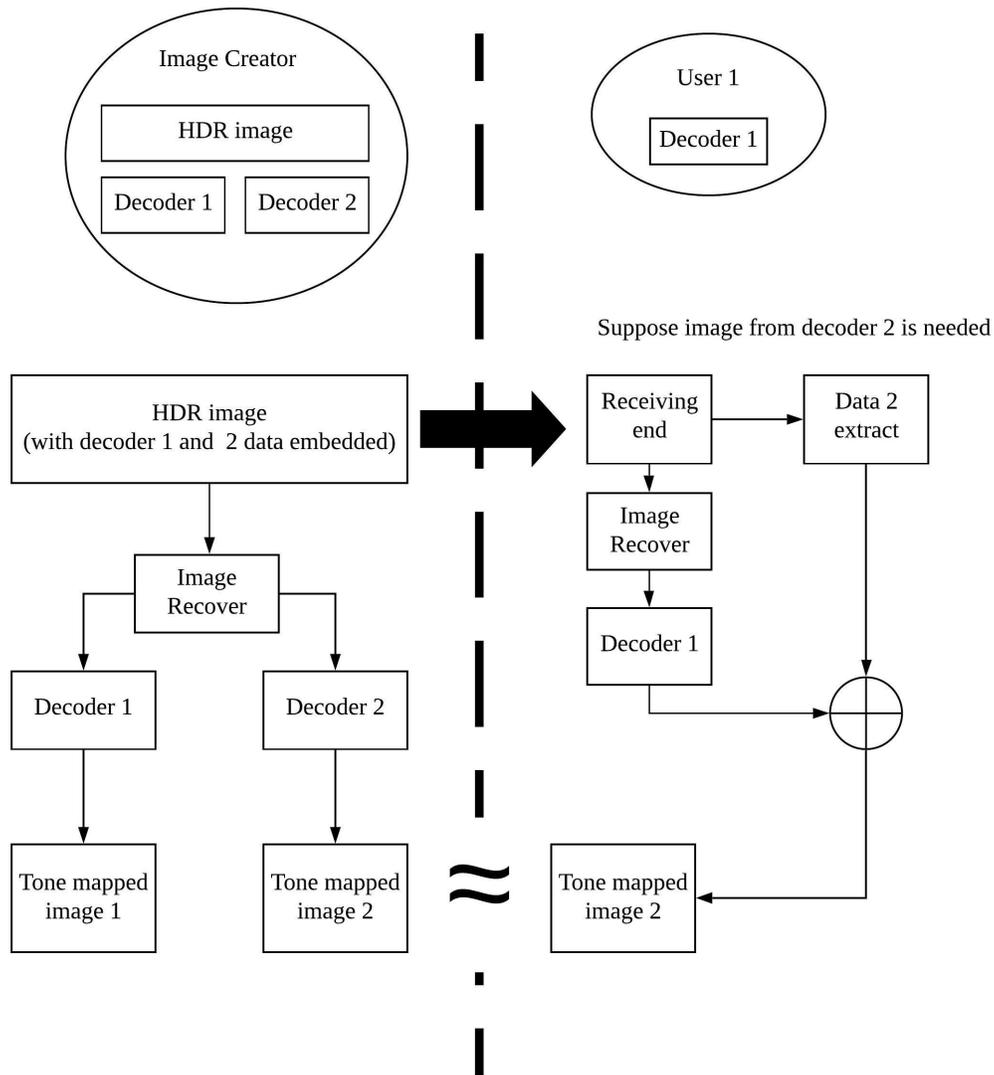


Figure 3.3.1: Data hiding application scenario for tone mapping

the DC coefficient bits and Table 3.6 shows the space (number of bits) required to store the DC coefficients for six example images. The average bit per pixels required will be 0.0359bpp, which is a volume that can be sufficiently accommodated by the proposed method. Further improvement of the proposed method will be explored to decrease the size of image difference D so that more image information can be embedded and image Z can be restored closer to the original X .

Table 3.6: Bits required to store the DC component of the difference image Δ .

Filename	Number of DC coeff	Bits per coefficient	Total bits
507	11926	2	23852
BalancedRock	11926	2	23852
AhwahneeGreatLounge	11926	2	23852
BandonSunset(1)	10184	3	30552
CemeteryTree(2)	11926	2	23852
Frontier	11926	3	35778

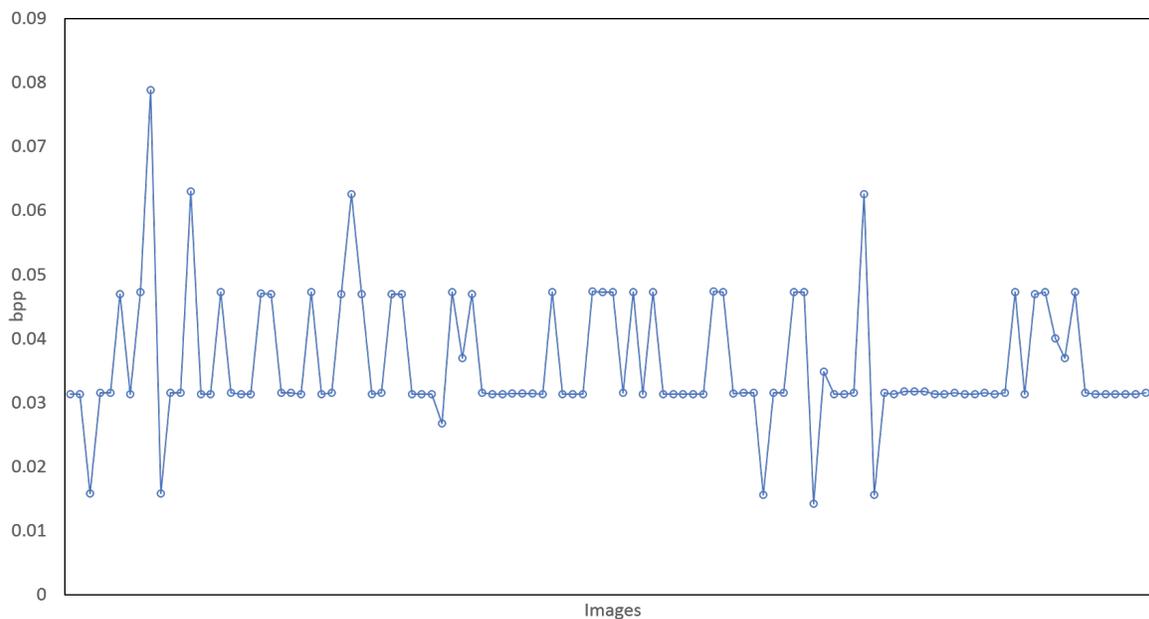


Figure 3.3.2: Capacity required to hide DC coefficient of image difference.

Chapter 4

Data Hiding in Masked OpenEXR image

In this chapter, a technique is put forward to first perceptually mask a HDR image, and then hide data into the masked HDR image. Specifically, pixels in the OpenEXR file, which are stored in half floating point precision format, are manipulated. Specifically, a predictor is utilized to predict pixel values, where well predicted pixel locations are flagged and utilized as the venues to hide data. On the other hand, the ill predicted pixel locations are flagged as unusable. Next, each pixel is divided into segments of 5 bits, and the segments are XOR-ed and permuted to mask the perceptual semantic of the HDR image. Data hiding then takes place at locations flagged as usable, which further distorts the quality of the image. The proposed method is both reversible and separable.

4.1 Proposed method

Based in the discussion in OpenEXR in Section 2.2.3, the exponent part of a pixel determines its coarse value, therefore the exponents of a natural image are highly correlated, particularly for a group of pixels in the same region. This trait of OpenEXR can be exploited for reversible data hiding, where the exponent values can be replaced by data bits but at the same time they can be completely recovered later through prediction. In addition, the altered exponent values can result in drastic change to the pixel values, causing severe quality degradation. Figure 4.1.1 shows the flow of processes in achieving perceptual masking and data hiding.

4.1.1 Preprocessing

In the first stage, the exponent part of all pixels are predicted by using a pixel predictor. The purpose of this step is to determine well predictable locations, so that they can be replaced by the data to be hidden. In this work, the Median Edge Detection (MED) predictor from JPEG-LS [114] is considered in this proposed method. MED is a simple predictor which relies on the direction of the edges, i.e., horizontal, vertical, or no edges (as discussed in Section 3.2).

The original exponent values E_o are treated as a 5-bit values, ranging from 0 to 31. The predicted value, denoted by E_p , is then computed by using MED, and the prediction error $e_p = E_o - E_p$ is computed. To flag the usability of each exponent for data hiding purpose, the value e_p is analyzed. Specifically, if $e_p = 0$ (viz., the prediction is perfect), the least significant bit (LSB) of E_o is modified to ‘0’. Otherwise, the LSB of E_o is set to ‘1’. For both cases, the remaining 4 bits remained unchanged. For the ill predicted exponent E_o , their original LSB are stored in a dynamic array C . However, for the well predicted exponent E_o , their LSB are not stored because the entire E_o can

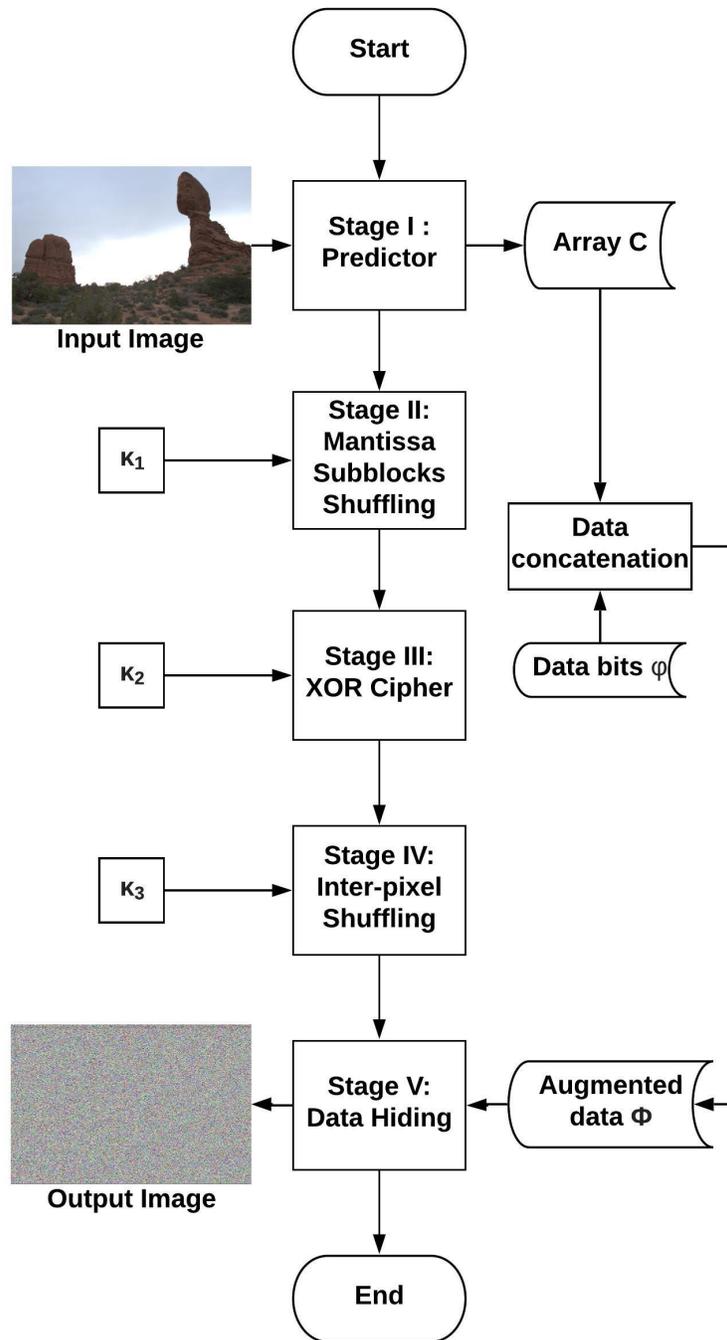


Figure 4.1.1: Overall process of proposed method.

be predicted perfectly. Note that here, the array C is important to achieve reversibility. Let E'_o denoted the modified exponent value, where $|E_o - E'_o| \leq 1$ since only the LSB is potentially changed. The LSB of E'_o then guides the data hiding process, where ‘0’ denotes that data can be hidden (or from the perspective of the receiver - data is hidden), while ‘1’ means that no data is hidden.

4.1.2 Masking The Image

First, the 10-bit mantissa part M is divided and treated as two independent blocks each with 5 bits, namely, M_1 and M_2 . The subblocks M_1 and M_2 each assumes a size of 5 bits, which is the same as the length of E_o to ease the masking operations. Therefore, each pixel will assume the form of $[S, E'_o, M_1, M_2]$, but for the purpose for this work, S is ignored. Second, M_1 and M_2 are extracted from all pixels, and shuffled by using a key κ_1 . The shuffled M_1 and M_2 form new mantissas (see Fig. 4.1.2). The potential outcomes for each pixel include:

$$\begin{aligned}
 & [E'_o(i), M_1(j), M_2(k)], \\
 & [E'_o(i), M_2(j), M_1(k)], \\
 & [E'_o(i), M_1(j), M_1(k)], j \neq k, \text{ or,} \\
 & [E'_o(i), M_2(j), M_2(k)], j \neq k,
 \end{aligned} \tag{4.1}$$

where $M_1(j)$ denotes the M_1 block from the j -th pixel, and $M_2(k)$ as well as $E'_o(i)$ are defined in same manners. The purpose of Stage II is to mask the details or textures (particularly recurring patterns) captured by the mantissa part, although mantissa can only cause small variation of value in comparison to the exponent part.

Next, the 4 most significant bits (4MSB) of the exponents E'_o for each pixel as well as its newly formed mantissa are XOR-ed with a random sequence of 0’s and

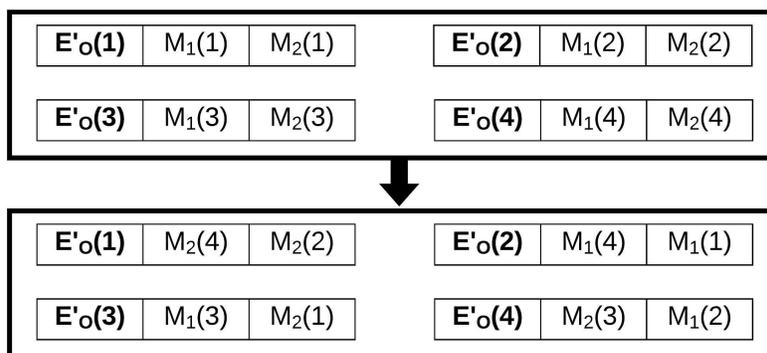


Figure 4.1.2: An example output of mixing M_1 and M_2 blocks among pixels (Stage II).

1's (e.g., stream cipher), which is generated with the key κ_2 (Stage III). Figure 4.1.3 illustrates an example, where E'_o and M'' refer to the modified exponent and mantissa parts, respectively. Note the LSB of E'_o does not undergo the XOR step to maintain the flag, i.e., $LSB(E'_o) = LSB(E''_o)$. This XOR step is important to break the high correlation between the pixels, particularly the ill predicted pixels, as they will not be involve the data hiding process in Stage V (see Section 4.1.3 for detailed explanation). The newly formed pixels are then shuffled by using the key κ_3 (Stage IV) to further increase the overall entropy of the image pixels.

It is noteworthy that Stages II, III and IV are independent to each other. Therefore, they can be implemented without following any particular order. In addition, a master key κ_m for content masking can be considered to generate the keys κ_1 , κ_2 and κ_3 through some scheduling algorithm.

4.1.3 Data Hiding

Let ϕ be the external data to be hidden, which is usually encrypted with a key κ_4 . For reversibility purpose, the original LSB of the exponent, denoted by C needs to be stored. Therefore, the augmented data $\Phi = [C|\phi]$ is constructed, where $|$ refers to the concatenation operation. Φ is then processed in segments of 4-bit. Data hiding

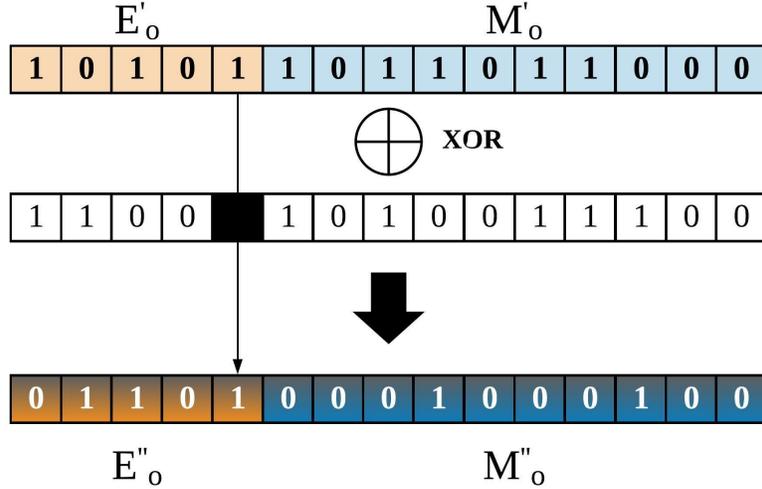


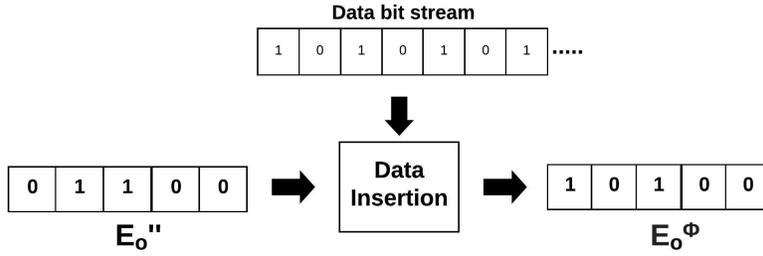
Figure 4.1.3: Masking pixel using XOR operation (Stage III).

is carried out by considering pixel locations where $LSB(E''_o) = 0$ for E''_o being the exponent part of the pixel at Stage IV. Specifically, for each usable pixel, the most significant 4 bits of its exponent are replaced by a 4-bit segment from Φ . Data hiding will not take place if $LSB(E''_o) = 1$. The process is repeated until all segments of Φ are inserted or when one runs out of usable pixels. Figure 4.1.4 illustrates the data insertion process, where the new exponent value is denoted as E_o^Φ .

4.1.4 Data Extraction and Image Recovery

The data extraction and image recovery steps are exactly the reverse of the encryption and data insertion processes. Specifically, the inserted data, i.e., Φ is first extracted, and it is processed to obtain ϕ and C . The intended receiver will be able to decrypt ϕ using κ_4 . Note that the data can be extracted without the need to decrypt the image.

To recover the original image, the encryption processes need to be reversed, i.e., inverse shuffling in Stage IV using κ_3 , XOR operation in Stage III using the same binary sequence generated by κ_2 , and inverse shuffling of the M_1 's and M_2 ' blocks in



(a) Case 1: E_o'' with LSB of '0'.



(b) Case 2: E_o'' with LSB of '1'. The bits are unchanged

Figure 4.1.4: Hiding data into preprocessed exponent part.

Stage II using κ_1 . Next, the LSB of the exponent part for each pixel is examined to reconstruct the image. Specifically, when $\text{LSB}(E) = '1'$, that means 4MSB planes were not replaced, i.e., the original sequence is intact, and the first bit of C is removed and used as the LSB for E . On the other hand, if $\text{LSB}(E) = '0'$, then 4MSB planes are recovered through prediction. Note that the predict-to-recover process must be carried out in raster scan order again to ensure the bits from C is reinserted at the correct position, as only pixels flagged as unpredictable have their original LSB stored in C (length of array $C \neq$ total pixels in image).

4.2 Experiments and Evaluations

The proposed algorithm is implemented in Matlab R2020a. 108 images from the dataset [21] are used for experiment purpose. Since the original image in [21] is large in dimension (i.e., larger than 4K), each test image is down-scaled to about 712×1072

pixels (which is similar to the experiment conditions in Section 3.1.1). Few example of these images are shown in Figure 3.1.4. It is confirmed that the inserted data can be extracted, and the original image can be completely reconstructed from its processed counterpart.

4.2.1 Effective Payload

First, the raw capacity achieved for each image using the proposed algorithm is computed as:

$$\frac{\textit{number of usable pixels}}{\textit{total pixels}} \times 100\%, \quad (4.2)$$

while the percentage of side information is computed as:

$$\frac{\textit{number of unusable pixels}}{\textit{number of usable pixels}} \times 100\%. \quad (4.3)$$

Here, raw capacity refers to the actual number of pixels that are flagged as suitable for hiding data, while side information percentage is the fraction of raw capacity spent on coding the array C . Capacity is the average number of bits can be hidden in each pixel.

Figure 4.2.1 shows the graph of capacity of 108 test images. The average capacity achieved is 3.244bpp. Table 4.1 records the results for six representative test image. Results suggest that the percentage of usable pixels is high for all HDR images considered, with an average of 81.1045%. Some images yield lower capacity because there are less well predictable pixels. In other words, higher number of ill predictable pixels result in relatively larger side information C , and hence less available space to insert data.

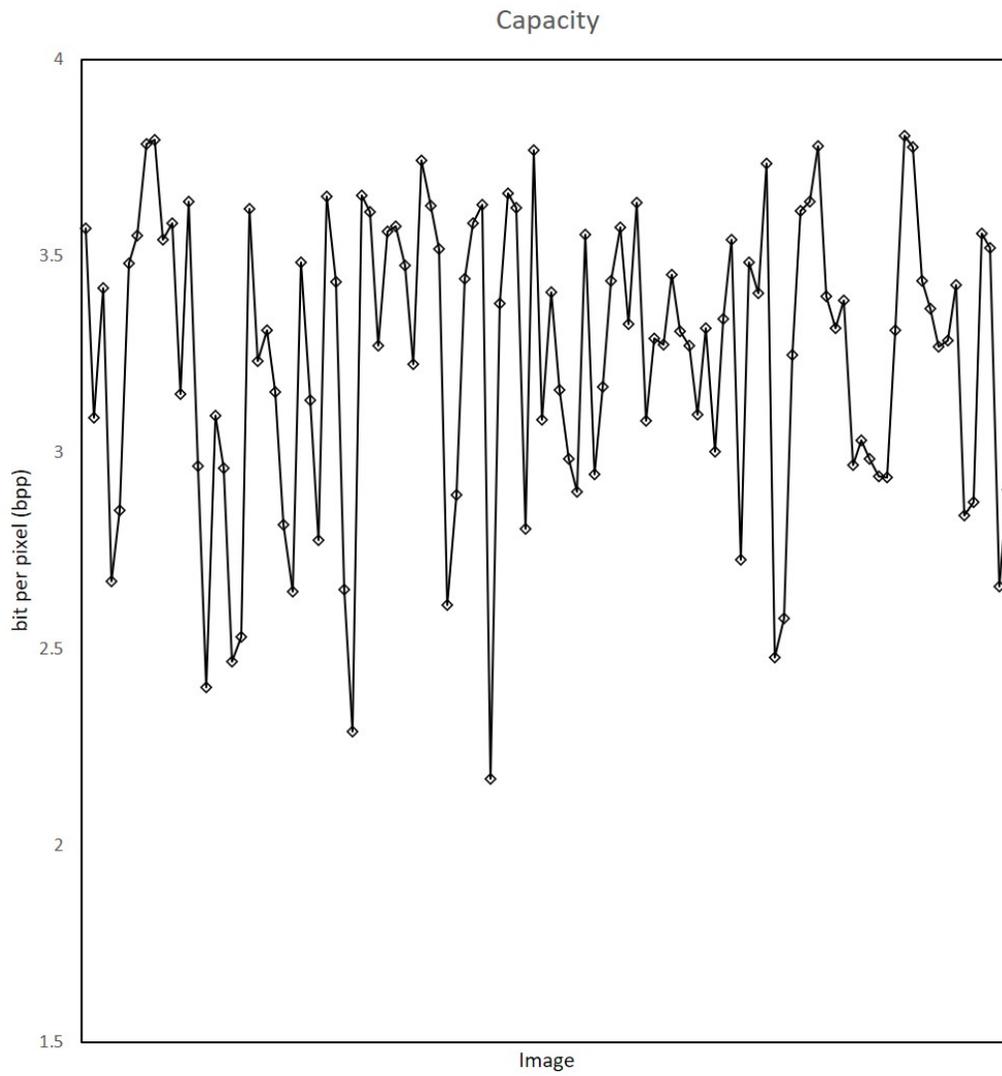


Figure 4.2.1: Graph of capacity of each test images

Table 4.1: Example hiding capacity for six representative test images.

Image	raw capacity (%)	side info capacity (%)	capacity (bpp)
507	89.2549	2.9442	3.5704
BalancedRock	88.5727	3.1595	3.5506
AhwahneeGreatLounge	77.2292	7.2955	3.0892
BandonSunset(1)	94.6154	1.3543	3.7846
CemeteryTree(2)	63.5323	14.2582	2.5300
Frontier	86.3199	3.8944	3.4347

4.2.2 Image Quality

This section presents the quality of the processed image from Stage V in two scenarios: the first one hides data after the prediction stage by skipping Stage II, III and IV, and; the second one hides data after the encryption stage, viz., stepping through Stage II, III and IV. By visual inspection, output image in the first scenario is already severely distorted. However, some details of the image are still visible (e.g., see outline of wheel at the lower left region in Fig. 4.2.3c). Nonetheless, the image produced in the second scenario is completely distorted, i.e., there is no trace of the original perceptual semantic. This observation verifies that although replacing the exponent part with payload bits already results in a highly distorted image, Stage II, III and IV are crucial to further intensify distortion to completely mask the perceptual semantic of an image. Figure 4.2.3 compares the representative output images for both scenarios.

To quantify distortion, log-based PSNR [58] is computed for the processed images for both scenarios, and the results are recorded in Figure 4.2.2. Table 4.2 shows the example results of six test images. As expected, the logPSNR values are low (i.e., close to 0) and this confirms that the processed images are highly distorted after applying the proposed image masking and data hiding technique. The values of logPSNR in Scenario 2 is lower than that of Scenario 1, which shows that the distortion is higher

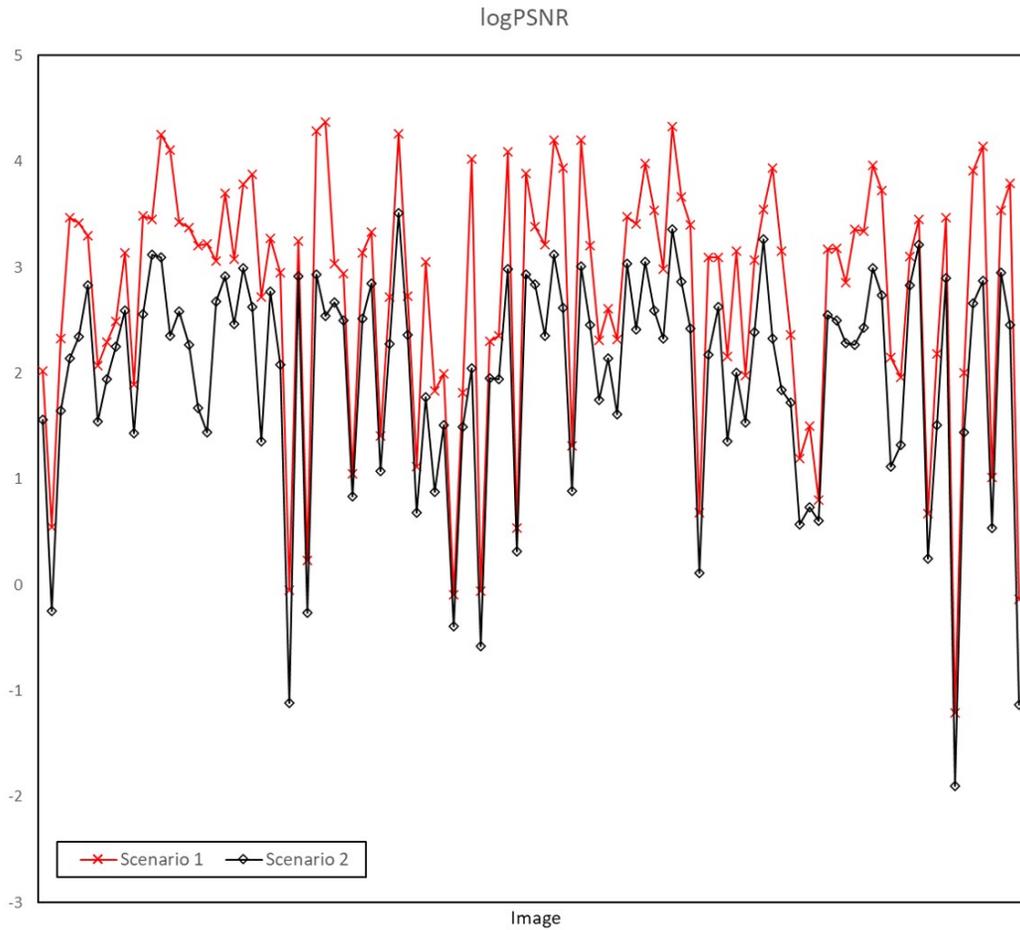


Figure 4.2.2: Graph of image quality assessment (logPSNR)

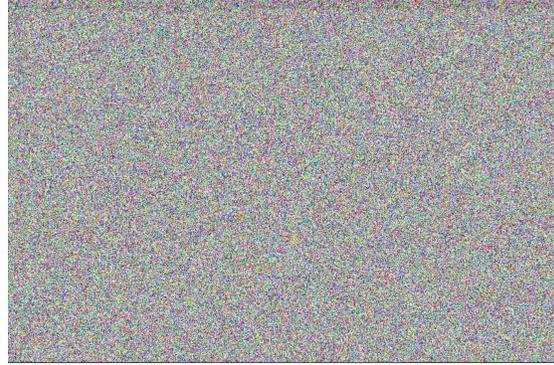
in Scenario 2.

4.2.3 Randomness test

To analyze the proposed method from the security aspect, the randomness test, namely, number of pixels change rate (NPCR) [116], is applied to the output images. The NPCR value ranges from 0 to 1. NPCR is designed to test the number of changed pixels between the masked images when the plain images are subtly modified. NPCR is commonly used to test the resistance of a cipher with respect to differential attacks.



(a) BalancedRock: Data hidden after Stage I (Scenario 1)



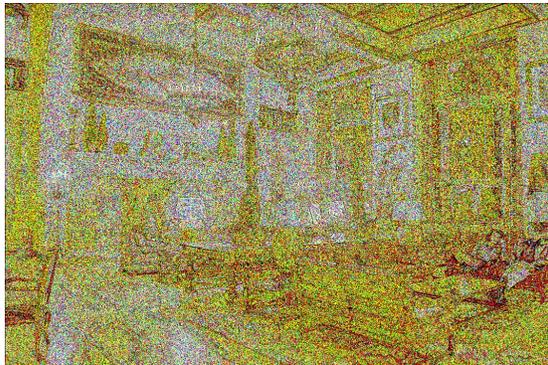
(b) BalancedRock: Data hidden after Stage IV (Scenario 2)



(c) 507: Data hidden after Stage I (Scenario 1)



(d) 507: Data hidden after Stage IV (Scenario 2)



(e) AhwahneeGreatLounge: Data hidden after Stage I (Scenario 1)



(f) AhwahneeGreatLounge: Data hidden after Stage IV (Scenario 2)

Figure 4.2.3: Output images for both scenario.

Table 4.2: Quality assessment after processing (dB).

Image	Log PSNR	
	Scenario 1	Scenario 2
507	3.3252	2.7679
CanadianFalls	4.4658	3.0105
CemeteryTree(2)	4.8320	2.7357
Frontier	3.3314	2.5717
BigfootPass	3.9878	3.4865
BalancedRock	3.4036	2.7252

The general equation of NPCR is:

$$NPCR : N(C^1, C^2) = \sum_{i,j} \frac{D(i,j)}{T}, \quad (4.4)$$

where

$$D = \begin{cases} 1, & \text{if } C^i(i, j) \neq C^2(i, j) \\ 0, & \text{if } C^i(i, j) = C^2(i, j) \end{cases}. \quad (4.5)$$

Here, C^1 and C^2 represent images before and after changing the pixel value. T represents the total number of pixels in the image. In this experiment, $C^1(i, j)$ is the pixel value of the original image and $C^2(i, j)$ is the pixel value of the marked image. After obtaining the value of D with Eq. (4.5), the NPCR value of the marked image is calculated with Eq. (4.5). Figure 4.2.4 shows the results of each test image. The resulting images can achieve high NPCR value, i.e., > 0.99 , which is desirable.

4.2.4 Key sensitivity test

The main purpose of key sensitivity test is to test how sensitive the proposed algorithm is with respect to the key used in masking the images. In the experiment, a masked image is generated by using key A . Then, key A is modified by one bit and the

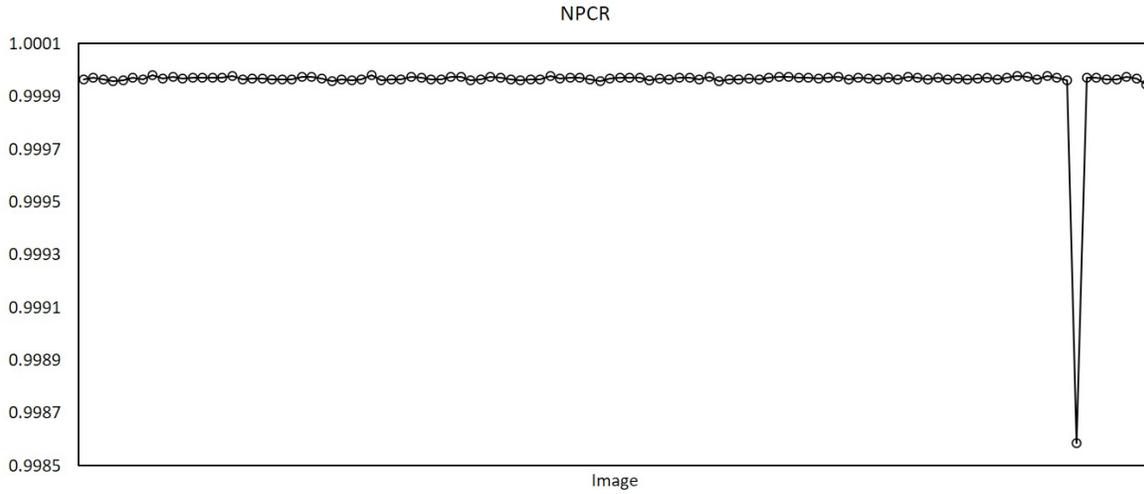


Figure 4.2.4: Number of pixels change rate (NPCR)

resulting key is denoted as B . The key B is then used to unmask the masked image and generate the image W . Finally, the quality of W is measured by using log-based PSNR [58]. The graph of logPSNR versus image is shown in Figure 4.2.5. It is observed that the logPSNR values are low with an average value of 2.6372, suggesting that the image cannot be unmasked with a wrong key, even when the difference is only one bit. Figure 4.2.6 shows some examples of the resulting images when using key A and key B for unmasking. It can be observed that the images generated from key B are highly distorted. Notice the unnatural bands at the bottom of the unmasked images. The reason for these unnatural bands to appear in the image W is due to the abrupt stop during the predict-to-recover process. This is caused by the incorrect prediction of the wrongly decoded pixel in the previous stage. As the number of bits in the array C extracted at the beginning of the data extraction process is the same as the number of ill-predicted pixels, the incorrect prediction will lead to insufficient number of bits from array C to be reinserted to the LSBs of the exponent values. Hence the decoding process stops when the array C is fully used. Based on this experiment, it can be concluded that an approximated key cannot unmask the original image, thus suggesting the sensitivity

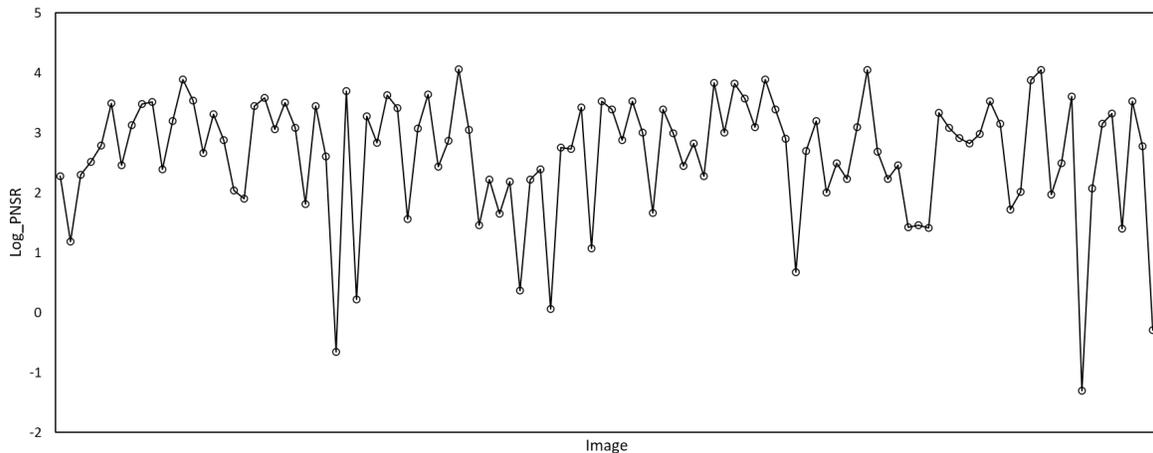


Figure 4.2.5: Graph of logPSNR values of unmasked images from key B . The average value of the logPSNR values is 2.6372.

of the proposed method on the key. In other words, only the exact key can unmask the image perfectly.

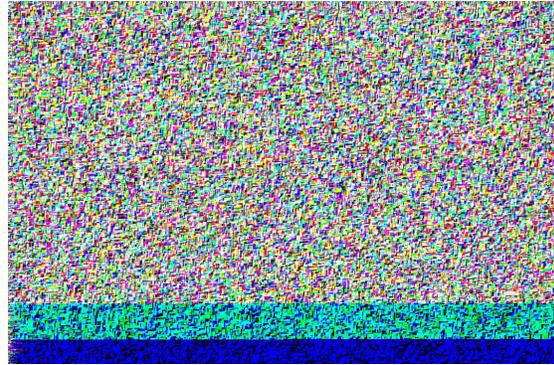
4.3 Discussion

In the proposed method, it is possible to extract the data hidden without recovering the image, i.e., direct extraction in Stage IV. The original position of the pixel blocks or pixels are not required for data extraction. However, the pixels position need to be recovered if and only if the recovery of original image is required. For the purpose of reversibility (i.e., ability to recover to original image), side information from the original image is needed to reconstructed the image. In the case when reversibility criteria is not needed, side information can be omitted during the data hiding process to maximize capacity for actual data bits.

Furthermore, the proposed method is suitable to be applied for floating point numbers, specifically half precision format. The conventional methods [106–108] are not suitable for floating point numbers because they are designed for integer pixels, which



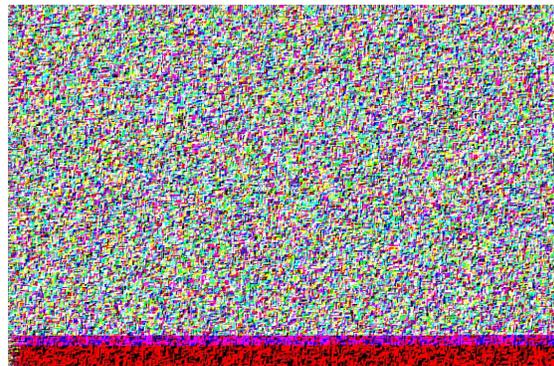
(a) BalancedRock:unmasked using key A



(b) BalancedRock:unmasked using key B



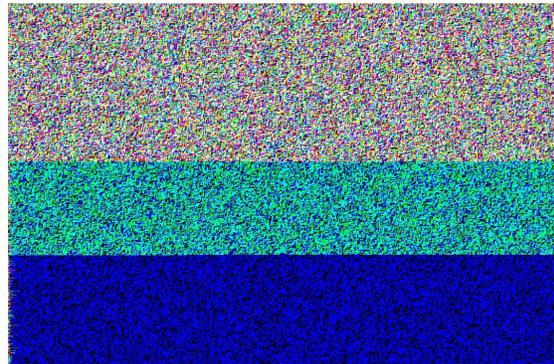
(c) 507: unmasked using key A



(d) 507: unmasked using key B



(e) AhwahneeGreatLounge: unmasked using key A



(f) AhwahneeGreatLounge: unmasked using key B

Figure 4.2.6: Output images when unmasking using the correct key (key A) and the slight modified key (key B)

have different intrinsic properties. For example, 3LSB are manipulated to hide data in [106], but the technique is not readily applicable to pixels stored in floating point format. Specifically, the mantissa part of the pixel in half-precision format is rather random and it is challenging to predict them precisely.

At the time of this writing, there are no other similar hide-to-mask methods for the OpenEXR file format. The most relevant one is work proposed by Ting *et al.* [117], which operates in the compressed domain based on the JPEG XT file format. In Ting *et al.*'s method, the capacity is fixed at 1/64 bits per pixel due to the nature of the size of the Minimum Coded Unit (MCU) block (i.e., 8×8). Depending on the bit to be hidden, the MCU block in the base layer and the residual layer are swapped to hide data. Here, the number of nonzero coefficients dictates whether the swapping of MCU blocks have occurred. More intensified image masking is achieved when the DC components are also swapped. On the other hand, for the method proposed in Section 4.1, the capacity is greatly dependent on the characteristic of the image. A pixel predictor is used to guide the embedding process, where only predictable pixels are utilized to hide data. Data is then hidden into the image while masking it by overwriting the exponent bits.

4.4 Summary and conclusion

In this chapter, image masking and data insertion techniques are put forward for HDR image stored in the OpenEXR format. Specifically, the exponent part of each pixel is classified as predictable or not predictable. The predictable pixels are modified further to hide data, while side information is stored as part of the payload for perfect image recovery purpose. To further intensify the distortion, the mantissa subblocks are processed by shuffling, XOR cipher and inter-pixel shuffling. Experimental results suggest that, on average, the number of usable pixels is greater than 70% of the total number

of pixels. Results also suggest that the quality of an image can be completely distorted and perfectly recovered later when required. The NPCR values show that the image is resistance to differential attack and the key sensitivity test shows that the proposed method is sensitive to the key used.

As future work, more detailed analysis will be performed to investigate into the security aspect of the proposed method. In addition, the file structure of OpenEXR will be further explored to improve capacity and secrecy.

Chapter 5

Scalable Coding for Backward Compatibility

HDR image require specific display devices to be rendered and displayed correctly. However, HDR display device is expensive for consumer use, and most display devices in use are still in LDR. On the other hand, most existing digital media are in LDR and the hardware capability of HDR display is not fully utilized when displaying these older contents. Therefore, method which can offer backward compatibility feature for LDR device, and vice versa, is needed. One of the methods which can achieves this feature is scalable coding. In this chapter, a scalable coding method for the OpenEXR file format is proposed. Specifically, the unique structure of the floating point pixel representation in OpenEXR image is exploited in the proposed method.

5.1 Proposed method

To achieve scalable coding, the coding structure of OpenEXR pixel, i.e., floating point value, is analyzed. Here, the mantissa part, which contains the details of an image,

is exploited for this purpose. This is because the detailed appearances (e.g., precise contour or detailed shading) are only visible through HDR device, but not available through LDR device. With this observation, a scalable coding method is put forward to package the mantissa component (basically rearrangement of the bit planes) to cater for various bit depth requirements.

Earlier in Figure 3.1.1, it is shown that the HDR image without the mantissa component (i.e., setting them to zero) still resembles its original counterpart, but HDR without the exponent component is hardly recognizable, let alone making the connection to the original image. Furthermore, it is also observed that most values in all images considered are positive, while only a small fraction of the pixel values are negative. To investigate into the contribution of sign component to the image quality, the image quality after the removal of the negative sign (i.e., essentially making all pixel values positive) is computed. The average quality (HDR-VDP-2) of 108 HDR images is 90.08, which is high, hence suggesting the high similarity before and after the removal of the negative sign. For visual inspection, the results for six representative HDR images are shown in Figure 5.1.1. Therefore, the sign component is given a lower priority in the coding order.

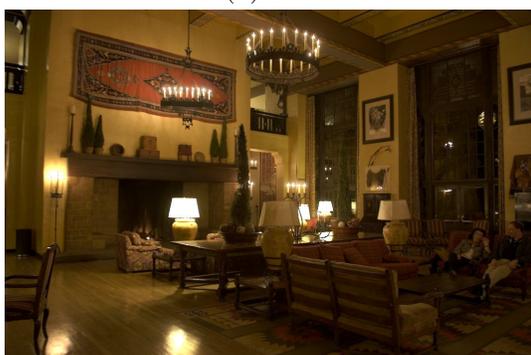
Based on the discussion above, it is concluded the exponent is the most important component, followed by the mantissa and sign bit. To facility image decoding in older systems, an OpenEXR file is split into 2 files. One may imagine that each 16-bit array (i.e., one floating point value) is split into 2 sub-arrays each encoding 8 bits, while the bits are also rearranged to facilitate direct decoding. In the extreme case of binary image, only the first bit (i.e., MSB) of the exponent is decoded. Hence, the pixel value will either be 0 or 1, which can be easily translated to 0 and 255 for direct display. This is named Mode 1. Next, 2 MSB from the exponent component can be decoded, where the values ranges from 0 to 3. For a simple direct display, the decoded values in the



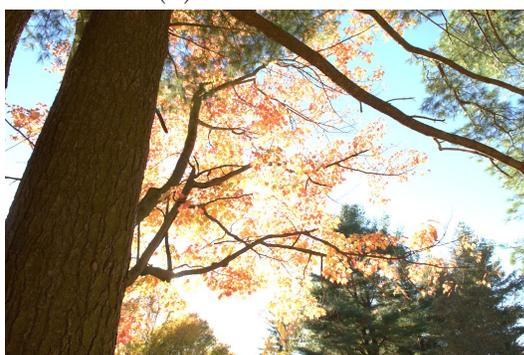
(a) 507



(b) BalancedRock



(c) AhwahneeGreatLounge



(d) CemeteryTree(2)



(e) Flamingo



(f) Frontier

Figure 5.1.1: Six representative HDR images with sign bit set to zero. Each image is tone-mapped for display purposes.

range [0,3] is scaled by 64. This approach is named Mode 2, and Mode 3, 4 and 5 are defined in a similar manner. However, the range of values in Mode 3, 4, 5 are [0, 7], [0, 15], and [0,31], respectively. The raw values are then scaled accordingly by multiplying 32, 16, and 8, sending them to the range of [0,255] for direct display purposes. (Note that, here, it is beyond the scope of this thesis to further improve the quality of these images through post processing techniques such as filtering).

In Mode 6, in addition to the exponent component (i.e., all 5 bits), the MSB of the mantissa component is included. Here, the values encoded are in the range of [0, 63], and they are multiplied by 4 for display purposes. Likewise, Mode 7 and 8 are defined in a similar manner by adding the 2MSB and 3MSB of the mantissa component, respectively. Here, the values decoded from mode 7 should be multiplied by 2 to send the values to the range of [0, 255]. It should be pointed out that the 8-bit array itself can be decoded directly by the decoder to render the image. This construct forms the first file, which can also be treated as an LDR version of the corresponding HDR content. As an example, the output images for each mode are shown in Figure 5.1.3 and Figure 5.1.4 for the images Frontier and Flamingo, respectively.

For the second file, it consists of 8 bits, including the remaining 7 bits from the mantissa, while the sign bit is included as the last component. An example of the arrangement of the bits in both the first and second files are shown in Figure 5.1.2. Note that the second file cannot be decoded by itself, since it contains information that is adding detailed textures to the image encoded by the first file. Nonetheless, to use any information encoded in the second file (even just one bit), the bit arrays from the first and second files need to be re-arranged based on their semantic, following the 16-bit floating point representation. Once the values are decoded, tone-mapping is performed so that the HDR (since it is dealing with more than 8 bits per pixel) image can be displayed properly on a LDR display device.

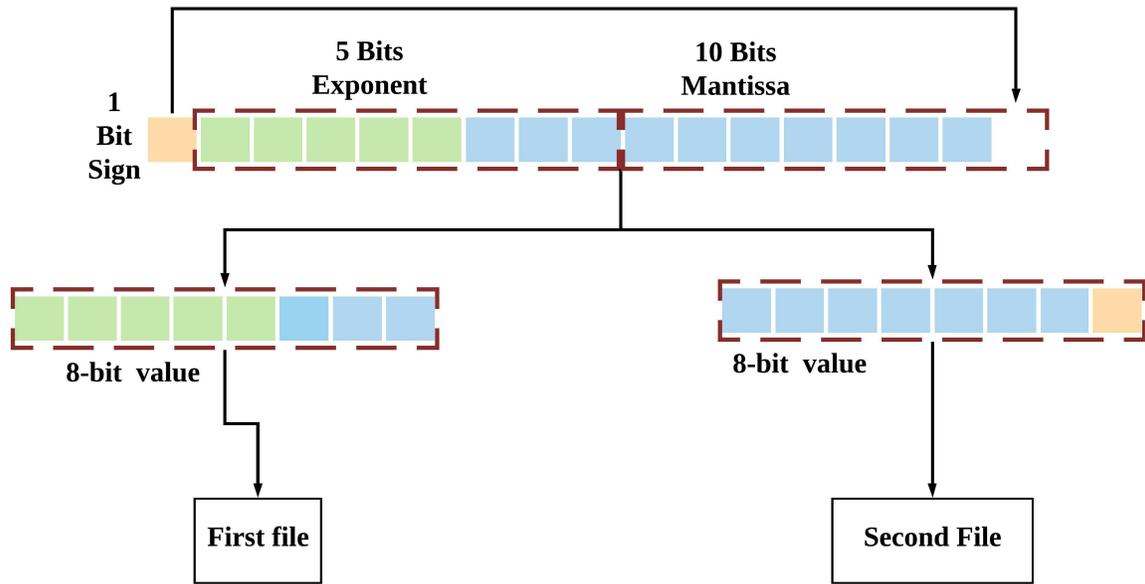
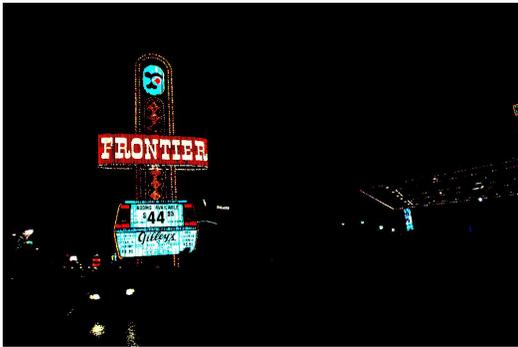


Figure 5.1.2: Arrangement of bits in both the first and second files (one byte each).

5.2 Results and Discussion

As seen from Figure 5.1.3 and Figure 5.1.4, the image quality improves as the number of bits used to form the image increases. In Mode 1, only bright pixels can be seen. As the mode number increases, more details can be seen in the generated image, and the images become more and more natural. In *Frontier* (see Figure 5.1.3), the banding in the sky becomes less and less visible until almost unnoticeable in Mode 8. In *Flamingo*, the increasing details visible as the mode number increases makes the building and the trees look more and more realistic. The slight distortion in images from the modes are not apparent when observed from a LDR device. Nonetheless, in general, the images appear to be washed out when viewed by directly treating the bit assemble as an integer. This is an expected outcome due to the exponential representation in 16-bit floating point value.

For demonstration purposes, an graphical user interface (GUI) is programmed in Matlab R2020a to showcase the scalable coding feature. A screenshot of the interface



(a) Mode 1



(b) Mode 2



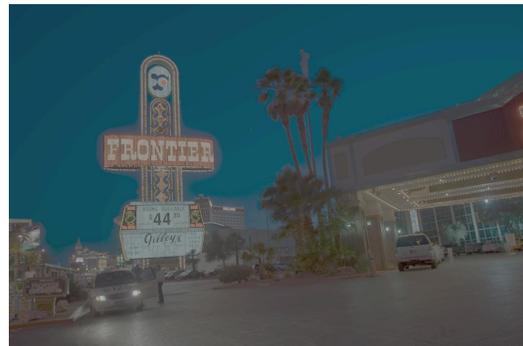
(c) Mode 3



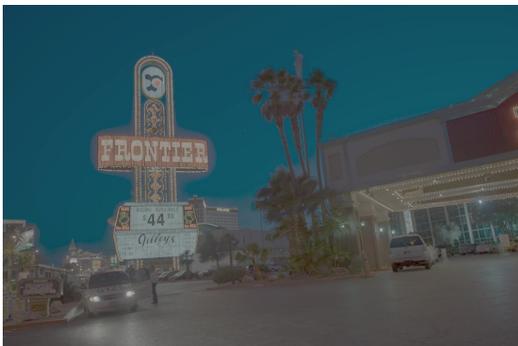
(d) Mode 4



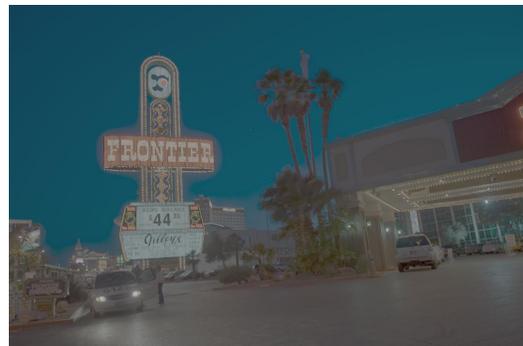
(e) Mode 5



(f) Mode 6



(g) Mode 7



(h) Mode 8

Figure 5.1.3: Different mode of display for *Frontier*



(a) Mode 1



(b) Mode 2



(c) Mode 3



(d) Mode 4



(e) Mode 5



(f) Mode 6



(g) Mode 7



(h) Mode 8

Figure 5.1.4: Different mode of display for *Flamingo*

is shown in Figure 5.2.1. The drop down box on the top enables the user to choose an OpenEXR image. The slider below select the number of bits in the mantissa to be included, where 0 means no mantissa bits are included and 10 means all mantissa bits are included. The interface shows the original image, the output image displayed as integer, output image displayed as floating point and the tone mapped version of the floating point image. Here, ‘integer’ refers to the process of treating the binary bits as integer number and displaying them directly by scaling them to 8-bit, similar to the different modes described above. On the other hand, ‘float’ refers to displaying the pixels in their original OpenEXR pixel format (viz., 16-bit floating point). Depending on the selected value from the slider, the LSBs that are excluded from displaying are set to zeros. Note that this GUI is operating in the LDR environment. From Figure 5.2.1, it can be observed that the resulting image that is handled as ‘float’ appears to be similar to the original image, as LDR environment cannot correctly rendered the original HDR image. The small distortion occur after modifying the bits will no be noticeable when display in LDR. When displaying as ‘integer’, pixels are treated as numbers in the range of $[0, 255]$, with constant step size. However, as discussed in Section 2.2.2, HDR usually adopts a logarithmic step size. Therefore, displaying the image with uniform step size causes the image to appear *washed out*.

The proposed scalable coding method for HDR image allows the rendering of an impression of its LDR counterpart directly on a LDR display device, without the need to perform tone mapping. Depending on the display capability of the device, different number of mantissa bits can be chosen (i.e., involving the first and second files), and the decoding operation changes depending on the number of bits to be decoded.

There is no other similar scalable coding **method** for the OpenEXR file format at the time of this writing. The most relevant work is the one proposed by Iwahashi *et al.* [20]. In Iwahashi *et al.*’s method, the method separates the HDR image into

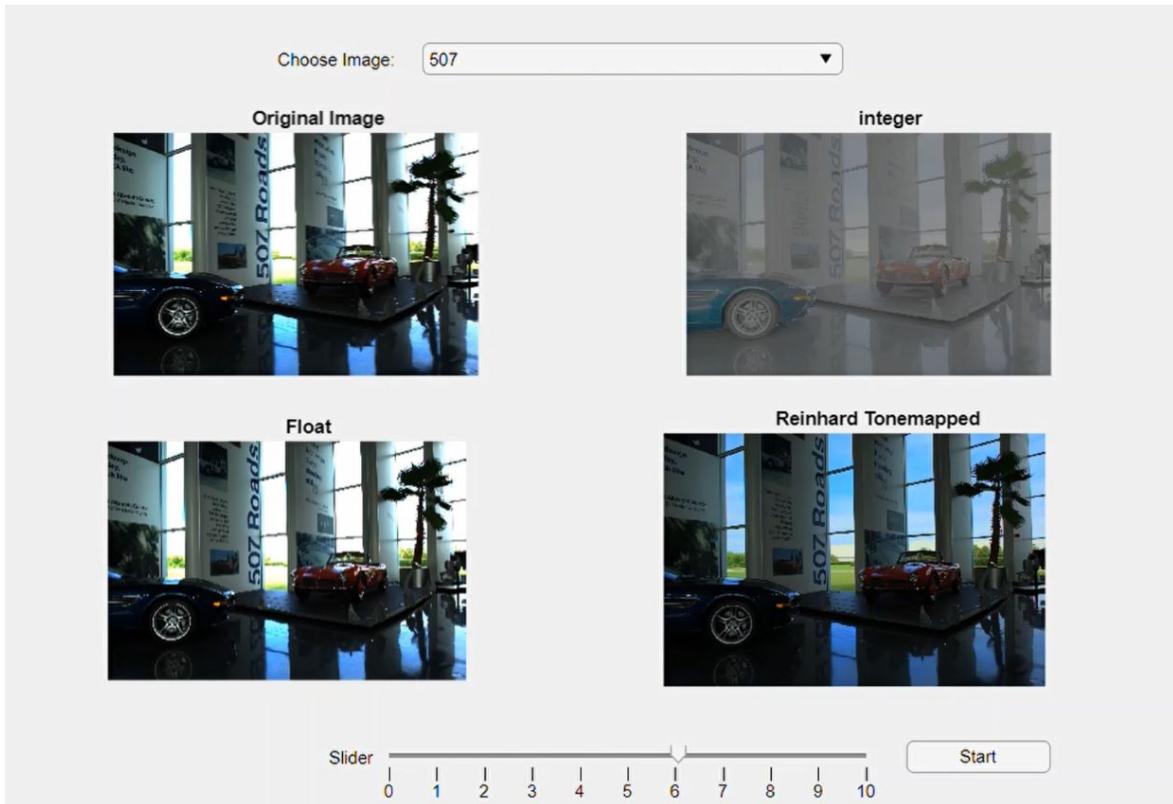


Figure 5.2.1: Scalable coding interface

two layers, namely, the base and enhancement layers. These layers are then converted to bitstream by using the JPEG2000 codec. The base layer is lossy coded while the enhancement layer is losslessly coded. When displaying the image, a tone mapping operator is used to convert the base layer to LDR. Their work aims to reduce the bit depth as well as the bit rate in the enhancement layer. On the other hand, for the method proposed in Section 5.1, the 5-bit exponent and the 3 MSB mantissa bits are grouped together and they form the first file. In addition, the 7 LSBs and the sign bit are grouped together and they form the second file. Although this somewhat resembles Iwahashi *et al.*'s method, it is simpler. The first file can be decoded directly without relying on any specific codec such as JPEG2000 and no TMO is involved. If needed, the second file can be decoded accordingly to access additional information to complete

the 16-bit float point information for each pixel location. There are also no modifications to bit depth.

5.3 Summary and Conclusions

In this chapter, a scalable coding method is designed to re-encode an OpenEXR HDR image by splitting the 16-bit array into two 8-bit arrays. The sign, exponent and mantissa components are rearranged to allow a direct decoding of the image without the need of performing tone mapping, although the directly decoded image appear to be washed out. It is observed that the quality of the LDR image gradually improves when more information is decoded. Specifically, the number of bits in the mantissa component to be included in the displaying process is manipulated to control the display quality. The lower the number of bits the lower the quality of the displayed image. However, the distortion is not noticeable in the LDR display devices.

As future work, a more efficient way to manipulate the mantissa component can be designed to further minimize the distortion in the image when allow direct displaying. In addition, scalable coding method which can exploit the relationship between the exponent and mantissa will also be further pursued.

Chapter 6

Conclusion

This thesis studied the emerging technology called high dynamic range imaging as well as the methods to protect them, specifically, data hiding and masking. Methods to ensure backward compatibility and their importance are also studied. In this chapter, a conclusion of this study is presented by summarizing the contributions made, as well as the limitations of the proposed methods. Finally, some future works and possible directions related to the proposed methods are suggested.

6.1 Summary

Two reversible data hiding methods are proposed specifically for the OpenEXR HDR file format by exploiting the properties of the half floating point format, which is the format adopted in OpenEXR to store the pixel values. Although the proposed methods can also be applied for the traditional applications such as fragile watermarking, hyper-linking related contents, etc., a new application of data hiding, i.e., to manage the display of the tone mapped image in different devices, is also proposed, which is unique to HDR image. Next, a reversible data hiding method in perceptually masked OpenEXR image is proposed to protect the images from unauthorized usage. Finally,

a simple scalable coding method designed for backward compatibility by re-encoding an OpenEXR image into two separate files, where one can be decoded directly without performing tone mapping.

6.2 Contributions

This study has achieved its objectives as follows:

1. Two data hiding methods are put forward to insert data into OpenEXR image. The first method (Section 3.1) converts the RGB pixel values to YCbCr and groups the values accordingly before applying histogram shifting method to reversibly hide data in the mantissa component. The second method (Section 3.2) utilizes a pixel predictor to categorize the pixels, indicating whether it is suitable for data hiding or not. Then the pixels undergo the second stage of selection based on their natural characteristics before inserting data bits into them. Both RDH methods can achieve high embedding capacity. Finally, in Section 3.3, a new application of data hiding in HDR image is put forward, aiming to encode a different impression of the tone-mapping LDR image.
2. A reversible data hiding method in perceptually masked OpenEXR image is proposed (Chapter 4). The proposed method first uses a pixel predictor (MED) to mark the pixels which are predictable. Then, the pixels undergo stages of modification to perceptually mask the visual semantics of the image. After masking the exponent part of the predictable pixels, they are replaced by the data bits to be embedded, which further distorts the image. Experiment results show that the proposed method can achieve high embedding capacity and at the same time masks the image completely.

3. A scalable coding method is proposed for OpenEXR image. Specifically, the bits (i.e., sign, exponent, and mantissa) in each pixels are rearranged to allow direct decoding on a LDR display device without performing tone mapping. The number of bits of the exponent and mantissa components determine the quality of image displayed, while sign bit was found to be having little influence. In addition, the distortion cause by LSBs in mantissa is found to be less noticeable when displaying in LDR display.

6.3 Limitations

Despite achieving encouraging results, this study has the following limitations:

1. The proposed methods are only applicable for the OpenEXR half floating point format pixels.
2. The reversible data hiding methods are designed solely for the purpose of inserting data into the HDR image without taking into account possible security attacks on the marked image, such as common watermark attacks including rotation, cropping, and compression.
3. The image masking method proposed did not include a key management scheme for Stage II to Stage IV.

6.4 Future research

For future research, the proposed methods can be improved to provide higher capacity as well as robustness against image processing techniques. Robust security features can also be explored for inclusion into the proposed methods. For example, in the first RDH method, when applying histogram shifting, the characteristic of the histograms

generated can be further analyzed to group the pixels more efficiently. A more detailed analysis can be performed to investigate the security aspect of the proposed methods, for both the RDH and image masking methods presented. Not only that, a more efficient way of adapting data hiding method for HDR display purpose can also be designed. In addition, the design of the scalable coding method can be further improved by considering the correlation between the exponent and mantissa components. Finally, more HDR file formats can be explored to achieve reversible data hiding and image masking. As each file format have their unique characteristic, in-depth analysis can be carried out on different HDR file formats to realise different applications of data hiding.

Bibliography

- [1] “LED display comes in the era of HDR,” <http://www.colorlight-led.com/new/led-display-comes-in-the-era-of-hdr.html>, [Accessed: 23-June-2021].
- [2] F. Banterle, A. Artusi, K. Debattista, and A. Chalmers, *Advanced High Dynamic Range Imaging, Second Edition*, 2nd ed. USA: A. K. Peters, Ltd., 2017.
- [3] “The Ins and Outs of HDR — What is HDR?” 2019, [Accessed: 2020-11-10]. [Online]. Available: <https://www.eizoglobal.com/library/management/ins-and-outs-of-hdr/index1.html>
- [4] R. K. Mantiuk, K. Myszkowski, and H.-P. Seidel, *High Dynamic Range Imaging*. American Cancer Society, 2015, pp. 1–42. [Online]. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1002/047134608X.W8265>
- [5] Kevin McCoy, “High-dynamic-range imaging — Wikipedia, the free encyclopedia,” https://en.wikipedia.org/wiki/High-dynamic-range_imaging, 2021, [Accessed: 15-April-2021].
- [6] K. Chia, K. Wong, and J.-L. Dugelay, “Data hiding in perceptually masked openxr image,” in *2019 IEEE 21st International Workshop on Multimedia Signal Processing (MMSP)*, 2019, pp. 1–6.

- [7] A. Artusi, R. K. Mantiuk, T. Richter, P. Hanhart, P. Korshunov, M. Agostinelli, A. Ten, and T. Ebrahimi, “Overview and evaluation of the JPEG XT HDR image compression standard,” *Journal of Real-Time Image Processing*, pp. 1–16, dec 2015. [Online]. Available: <http://link.springer.com/10.1007/s11554-015-0547-x>
- [8] Roshni Y, “Liquid Crystal Display (LCD),” <https://electronicsdesk.com/liquid-crystal-display.html>, 2021, [Accessed: 26-June-2021].
- [9] Tom Harris, “How Plasma Displays Work,” <https://electronics.howstuffworks.com/plasma-display.htm>, 2002, [Accessed: 26-June-2021].
- [10] “An introduction to OLED displays - What is an OLED?” <https://www.oled-info.com/oled-introduction>, 2021, [Accessed: 28-June-2021].
- [11] H. Seetzen, W. Heidrich, W. Stuerzlinger, G. Ward, L. Whitehead, M. Trentacoste, A. Ghosh, and A. Vorozcovs, “High dynamic range display systems,” in *ACM SIGGRAPH 2004 Papers*, ser. SIGGRAPH '04. New York, NY, USA: Association for Computing Machinery, 2004, p. 760–768. [Online]. Available: <https://doi-org.ezproxy.lib.monash.edu.au/10.1145/1186562.1015797>
- [12] “BrightSide DR37-P HDR display,” https://bit-tech.net/reviews/tech/brightside_hdr_edr/1/, 2005, [Accessed: 30-June-2021].
- [13] E. Reinhard, G. Ward, S. Pattanaik, and P. Debevec, *High Dynamic Range Imaging: Acquisition, Display, and Image-Based Lighting (The Morgan Kaufmann Series in Computer Graphics)*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2010.
- [14] Guorong Xuan, Jiang Zhu, Jidong Chen, Y. Q. Shi, Zhicheng Ni, and Wei Su, “Distortionless data hiding based on integer wavelet transform,” *Electronics Letters*, vol. 38, no. 25, pp. 1646–1648, 2002.

- [15] L. Xiong, Z. Xu, and Y.-Q. Shi, “An integer wavelet transform based scheme for reversible data hiding in encrypted images,” *Multidimensional Systems and Signal Processing*, vol. 29, pp. 1–12, 05 2017.
- [16] M. Fujiyoshi and H. Kiya, “A Blind Reversible Data Hiding Method for High Dynamic Range Images Taking Advantage of Sparse Histogram,” in *Digital Forensics and Watermarking: 16th International Workshop , IWDW 2017, Magdeburg, Germany, August 23-25, 2017, Proceedings*. Cham: Springer International Publishing, 2017, pp. 347–361. [Online]. Available: <https://doi.org/10.1007/978-3-319-64185-0{ }26>
- [17] Y. Bai, G. Jiang, Z. Zhu, H. Xu, and Y. Song, “Reversible data hiding scheme for high dynamic range images based on multiple prediction error expansion,” *Signal Processing: Image Communication*, vol. 91, p. 116084, 2021. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0923596520302113>
- [18] K.-S. Lin, T.-H. Chen, C.-H. Lin, and S.-S. Chang, “A tailor-made encryption scheme for high-dynamic range images,” in *Genetic and Evolutionary Computing*, J.-S. Pan, P. Krömer, and V. Snášel, Eds. Cham: Springer International Publishing, 2014, pp. 183–192.
- [19] G. Ward and M. Simmons, “JPEG-HDR: a backwards-compatible, high dynamic range extension to JPEG,” in *ACM SIGGRAPH 2005 Courses on - SIGGRAPH '05*. New York, New York, USA: ACM Press, 2005, p. 2. [Online]. Available: <http://portal.acm.org/citation.cfm?doid=1198555.1198708>
- [20] M. Iwahashi, T. Yoshida, N. B. Mokhtar, and H. Kiya, “Bit-depth scalable lossless coding for high dynamic range images,” *EURASIP Journal on Advances*

- in Signal Processing*, vol. 2015, no. 1, p. 22, dec 2015. [Online]. Available: <https://asp-eurasipjournals.springeropen.com/articles/10.1186/s13634-015-0209-y>
- [21] M. D. Fairchild, “The HDR Photographic Survey,” 2007. [Online]. Available: <http://rit-mcsl.org/fairchild/HDRPS/CIC15HDRSurvey.pdf>
- [22] R. C. Gonzalez and R. E. R. E. Woods, *Digital image processing*. Prentice Hall, 2008. [Online]. Available: <https://dl.acm.org/citation.cfm?id=1076432>
- [23] “Galaxy S21 5G | S21+ 5G,” <https://www.samsung.com/my/smartphones/galaxy-s21-5g/>, 2021, [Accessed: 12-April-2021].
- [24] “Mi 11,” <https://www.mi.com/global/mi-11/overview>, 2021, [Accessed: 12-April-2021].
- [25] Y.-Q. Shi, X. Li, X. Zhang, H.-T. Wu, and B. Ma, “Reversible data hiding: Advances in the past two decades,” *IEEE Access*, vol. 4, pp. 3210–3237, 2016. [Online]. Available: <http://ieeexplore.ieee.org/document/7479451/>
- [26] E. Reinhard, M. Stark, P. Shirley, and J. Ferwerda, “Photographic tone reproduction for digital images,” *ACM Trans. Graph.*, vol. 21, no. 3, pp. 267–276, Jul. 2002. [Online]. Available: <http://doi.acm.org/10.1145/566654.566575>
- [27] D. Mustafi, A. H. Engel, and K. Palczewski, “Structure of cone photoreceptors,” *Progress in Retinal and Eye Research*, vol. 28, no. 4, pp. 289–302, 2009. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1350946209000330>
- [28] C. Bloch, *The HDRI Handbook: High Dynamic Range Imaging for Photographers and CG Artists +DVD*. Rocky Nook, 2007.

- [29] K. Myszkowski, R. Mantiuk, and G. Krawczyk, "High dynamic range video," *Synthesis Lectures on Computer Graphics and Animation*, vol. 2, no. 1, pp. 1–158, 2008. [Online]. Available: <https://doi.org/10.2200/S00109ED1V01Y200806CGR005>
- [30] D. Hough, "Applications of the proposed ieee 754 standard for floating-point arithmetic," *Computer*, vol. 14, no. 3, pp. 70–74, 1981.
- [31] T. Richter, A. Artusi, and T. Ebrahimi, "JPEG XT: A New Family of JPEG Backward-Compatible Standards," *IEEE MultiMedia*, vol. 23, no. 3, pp. 80–88, jul 2016. [Online]. Available: <http://ieeexplore.ieee.org/document/7535096/>
- [32] F. Kainz, R. Bogart, P. Stanczyk, and P. Hillman, "Technical Introduction to OpenEXR," 2013. [Online]. Available: <http://www.openexr.com/TechnicalIntroduction.pdf>
- [33] G. Ward, "REAL PIXELS," *Graphics Gems II*, pp. 80–83, jan 1991. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/B9780080507545500256>
- [34] G. W. Larson and G. Ward, "LogLuv Encoding for Full-Gamut, High-Dynamic Range Images," *Journal of Graphics Tools*, vol. 3, no. 1, pp. 15–31, jan 1998. [Online]. Available: <http://www.tandfonline.com/doi/abs/10.1080/10867651.1998.10487485>
- [35] F. Banterle, A. Artusi, K. Debattista, and A. Chalmers, *Advanced high dynamic range imaging : theory and practice*. A K Peters, 2011. [Online]. Available: <https://dl.acm.org/citation.cfm?id=1996408>
- [36] "Ieee standard for floating-point arithmetic," *IEEE Std 754-2008*, pp. 1–70, Aug 2008.

- [37] A. Artusi, R. K. Mantiuk, T. Richter, P. Korshunov, P. Hanhart, T. Ebrahimi, and M. Agostinelli, “JPEG XT: A Compression Standard for HDR and WCG Images [Standards in a Nutshell],” *IEEE Signal Processing Magazine*, vol. 33, no. 2, pp. 118–124, mar 2016. [Online]. Available: <http://ieeexplore.ieee.org/document/7426553/>
- [38] *ISO/IEC 18477-3:2015 Information technology - Scalable compression and coding of continuous-tone still images - Part 3: Box file format*, 2015.
- [39] “OpenEXR,” <https://www.openexr.com/>, 2021, [Accessed: 23-June-2021].
- [40] C. Giardina, “Motion Picture Academy Expands its Software Foundation With New Projects, Members,” *The Hollywood Reporter*. [Online]. Available: <https://www.hollywoodreporter.com/movies/movie-news/motion-picture-academy-expands-software-foundation-new-projects-members-1206177/>
- [41] Theresa Knott, “Cathode-ray tube — Wikipedia, the free encyclopedia,” https://en.wikipedia.org/wiki/Cathode-ray_tube, 2021, [Accessed: 26-June-2021].
- [42] “What Is a CRT Monitor?” <https://www.easytechjunkie.com/what-is-a-crt-monitor.htm>, [Accessed: 26-June-2021].
- [43] E. Reinhard, G. Ward, S. Pattanaik, and P. Debevec, *High dynamic range imaging : acquisition, display, and image-based lighting*. Morgan Kaufmann, 2006.
- [44] W. Greenwald. With LG Out, Plasma HDTVs Are Dead. [Online]. Available: <https://www.pcmag.com/article2/0,2817,2471112,00.asp>
- [45] J. Minor. Panasonic Ends Plasma TV Production, Expands Tesla Battery Deal. [Online]. Available: <https://www.pcmag.com/article2/0,2817,2426629,00.asp>

- [46] S. Tew. Samsung to end plasma TV production this year. [Online]. Available: <https://www.cnet.com/news/samsung-reportedly-ending-plasma-tv-production/>
- [47] E. Devaney. How Does an LED TV Work? [Online]. Available: <https://smallbusiness.chron.com/led-tv-work-26798.html>
- [48] K. Wouk and P. Hall. OLED vs. LED: Which kind of TV display is better? [Online]. Available: <https://www.digitaltrends.com/home-theater/oled-vs-led/>
- [49] J. Carter. OLED vs LCD & LED: which TV technology should earn a place in your living room? [Online]. Available: <https://www.techradar.com/news/oled-vs-lcd-led>
- [50] “BrightSide Technologies — Wikipedia, the free encyclopedia,” https://en.wikipedia.org/wiki/BrightSide_Technologies, 2021, [Accessed: 29-June-2021].
- [51] “VESA DisplayHDR,” <https://displayhdr.org/>, [Accessed: 07-July-2021].
- [52] Z. Wang and A. Bovik, *Modern Image Quality Assessment*, 1st ed. Morgan & Claypool Publishers, 2006.
- [53] P. Hanhart, M. V. Bernardo, M. Pereira, A. M. G. Pinheiro, and T. Ebrahimi, “Benchmarking of objective quality metrics for hdr image quality assessment,” *EURASIP Journal on Image and Video Processing*, vol. 2015, no. 1, p. 39, Dec 2015. [Online]. Available: <https://doi.org/10.1186/s13640-015-0091-4>
- [54] K. M. Rafal Mantiuk, Scott J. Daly and H.-P. Seidel, “Predicting visible differences in high dynamic range images: model and its calibration,” 2005. [Online]. Available: <https://doi.org/10.1117/12.586757>

- [55] R. Mantiuk, K. J. Kim, A. G. Rempel, and W. Heidrich, “HDR-VDP-2,” *ACM Transactions on Graphics*, vol. 30, no. 4, p. 1, jul 2011. [Online]. Available: <http://portal.acm.org/citation.cfm?doid=2010324.1964935>
- [56] M. Narwaria, M. P. D. Silva, and P. L. Callet, “HDR-VQM: An objective quality measure for high dynamic range video,” *Signal Processing: Image Communication*, vol. 35, pp. 46 – 60, 2015. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0923596515000703>
- [57] V. C. Smith and J. Pokorny, “3 - color matching and color discrimination,” in *The Science of Color (Second Edition)*, second edition ed., S. K. Shevell, Ed. Amsterdam: Elsevier Science Ltd, 2003, pp. 103 – 148. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/B9780444512512500040>
- [58] T. O. Aydin, R. Mantiuk, and H.-P. Seidel, “Extending quality metrics to full luminance range images,” B. E. Rogowitz and T. N. Pappas, Eds., vol. 6806. International Society for Optics and Photonics, feb 2008, p. 68060B. [Online]. Available: <http://proceedings.spiedigitallibrary.org/proceeding.aspx?doi=10.1117/12.765095>
- [59] Zhou 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, 2004.
- [60] S. J. Daly, “Visible differences predictor: an algorithm for the assessment of image fidelity,” 1992. [Online]. Available: <https://doi.org/10.1117/12.135952>
- [61] F. Drago, K. Myszkowski, T. Annen, and N. Chiba, “Adaptive logarithmic mapping for displaying high contrast scenes,” *Computer Graphics Forum*, vol. 22,

- no. 3, pp. 419–426, 2003. [Online]. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1111/1467-8659.00689>
- [62] G. W. Larson, H. Rushmeier, and C. Piatko, “A visibility matching tone reproduction operator for high dynamic range scenes,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 3, no. 4, pp. 291–306, Oct 1997.
- [63] M. Ashikhmin, “A tone mapping algorithm for high contrast images,” in *Proceedings of the 13th Eurographics Workshop on Rendering*, ser. EGRW ’02. Aire-la-Ville, Switzerland, Switzerland: Eurographics Association, 2002, pp. 145–156. [Online]. Available: <http://dl.acm.org/citation.cfm?id=581896.581916>
- [64] F. Durand and J. Dorsey, “Fast bilateral filtering for the display of high-dynamic-range images,” *ACM Trans. Graph.*, vol. 21, no. 3, pp. 257–266, Jul. 2002. [Online]. Available: <http://doi.acm.org/10.1145/566654.566574>
- [65] R. Fattal, D. Lischinski, and M. Werman, “Gradient domain high dynamic range compression,” *ACM Trans. Graph.*, vol. 21, no. 3, pp. 249–256, Jul. 2002. [Online]. Available: <http://doi.acm.org/10.1145/566654.566573>
- [66] E. Reinhard, G. Ward, S. Pattanaik, and P. Debevec, *High Dynamic Range Imaging: Acquisition, Display, and Image-Based Lighting (The Morgan Kaufmann Series in Computer Graphics)*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2005.
- [67] H. Landis, “Production-ready global illumination,” 2002, pp. 87–101.
- [68] I. Cox, M. Miller, J. Bloom, J. Fridrich, and Kalker Ton, *Digital watermarking and steganography*. Morgan Kaufmann Publishers, 2008.

- [69] F. Y. Shih, *Digital watermarking and Steganography : Fundamentals and Techniques*. CRC Press, 2017.
- [70] C. Rey and J.-L. Dugelay, “Blind detection of malicious alterations on still images using robust watermarks,” in *IEE Seminar on Secure Images and Image Authentication (Ref. No. 2000/039)*, 2000, pp. 7/1–7/6.
- [71] I. Cox, M. Miller, J. Bloom, J. Fridrich, and Kalker Ton, *Digital watermarking and steganography*. Morgan Kaufmann Publishers, 2008.
- [72] L. H. Newman. You can send invisible messages with subtle font tweaks. [Online]. Available: <https://www.wired.com/story/fontcode-invisible-messages-steganography/>
- [73] J. Lahut and S. Cook. GE engineer charged with trade secret theft. [Online]. Available: <https://dailygazette.com/article/2018/08/01/feds-at-niskayuna-home-in-apparent-raid>
- [74] L. H. Newman. Mysterious ‘muslimcrypt’ app helps jihadists send covert messages. [Online]. Available: <https://www.wired.com/story/muslimcrypt-steganography/>
- [75] Z. Ni, Y.-Q. Shi, N. Ansari, and W. Su, “Reversible data hiding,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 16, no. 3, pp. 354–362, mar 2006.
- [76] Jun Tian, “Reversible data embedding using a difference expansion,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 8, pp. 890–896, 2003.

- [77] D. Thodi and J. Rodríguez, “Reversible Watermarking by Prediction-Error Expansion,” in *Image Analysis and Interpretation, 2004. 6th IEEE Southwest Symposium on*, no. 1. IEEE, 2004, pp. 21–25. [Online]. Available: <http://ieeexplore.ieee.org/document/1300937/>
- [78] S. Ong and K. Wong, “Rotational based rewritable data hiding in jpeg,” in *2013 Visual Communications and Image Processing (VCIP)*, 2013, pp. 1–6.
- [79] J. Ting, K. Wong, and S. Ong, “Rewritable data embedding in jpeg xt image using coefficient count across layers,” in *2019 IEEE 21st International Workshop on Multimedia Signal Processing (MMSP)*, 2019, pp. 1–6.
- [80] A. Alattar, “Reversible Watermark Using the Difference Expansion of a Generalized Integer Transform,” *IEEE Transactions on Image Processing*, vol. 13, no. 8, pp. 1147–1156, aug 2004. [Online]. Available: <http://ieeexplore.ieee.org/document/1315703/>
- [81] H. J. Kim, V. Sachnev, Y. Q. Shi, J. Nam, and H. G. Choo, “A novel difference expansion transform for reversible data embedding,” *IEEE Transactions on Information Forensics and Security*, vol. 3, no. 3, pp. 456–465, sep 2008. [Online]. Available: <http://ieeexplore.ieee.org/document/4539274/>
- [82] Lixin Luo, Zhenyong Chen, Ming Chen, Xiao Zeng, and Zhang Xiong, “Reversible Image Watermarking Using Interpolation Technique,” *IEEE Transactions on Information Forensics and Security*, vol. 5, no. 1, pp. 187–193, mar 2010. [Online]. Available: <http://ieeexplore.ieee.org/document/5313862/>
- [83] W. Hong, T.-S. Chen, and C.-W. Shiu, “Reversible data hiding for high quality images using modification of prediction errors,” *Journal of Systems and Software*, vol. 82, no. 11, pp. 1833–1842, nov 2009. [Online].

Available: <https://www-sciencedirect-com.ezproxy.lib.monash.edu.au/science/article/pii/S0164121209001204?via{%}3Dihub>

- [84] Xiaolong Li, Bin Yang, and Tieyong Zeng, “Efficient Reversible Watermarking Based on Adaptive Prediction-Error Expansion and Pixel Selection,” *IEEE Transactions on Image Processing*, vol. 20, no. 12, pp. 3524–3533, dec 2011. [Online]. Available: <http://ieeexplore.ieee.org/document/5762603/>
- [85] V. Sachnev, H. J. Kim, J. Nam, S. Suresh, and Y. Q. Shi, “Reversible watermarking algorithm using sorting and prediction,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 19, no. 7, pp. 989–999, 2009.
- [86] X. Li, J. Li, B. Li, and B. Yang, “High-fidelity reversible data hiding scheme based on pixel-value-ordering and prediction-error expansion,” *Signal Processing*, vol. 93, no. 1, pp. 198 – 205, 2013. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0165168412002551>
- [87] X. Qu and H. J. Kim, “Pixel-based pixel value ordering predictor for high-fidelity reversible data hiding,” *Signal Processing*, vol. 111, pp. 249 – 260, 2015. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0165168415000031>
- [88] S. A. Hiary and I. F. Jafar, “An efficient multi-predictor reversible data hiding algorithm based on performance evaluation of different prediction schemes,” *Multimedia Tools and Applications*, vol. 76, pp. 2131–2157, 2015.
- [89] C.-F. Lee, C.-Y. Weng, and C.-Y. Kao, “Reversible data hiding using lagrange interpolation for prediction-error expansion embedding,” *Soft Computing*, vol. 23, pp. 9719–9731, 2019.
- [90] G. Xuan, X. Tong, J. Teng, X. Zhang, and Y. Q. Shi, “Optimal histogram-pair and prediction-error based image reversible data hiding,” in *IWDW*, 2012.

- [91] H.-T. Wu and J. Huang, “Reversible image watermarking on prediction errors by efficient histogram modification,” *Signal Processing*, vol. 92, no. 12, pp. 3000 – 3009, 2012. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S016516841200179X>
- [92] J. Wang, J. Ni, X. Zhang, and Y. Shi, “Rate and distortion optimization for reversible data hiding using multiple histogram shifting,” *IEEE Transactions on Cybernetics*, vol. 47, no. 2, pp. 315–326, 2017.
- [93] Y. Jia, Z. Yin, X. Zhang, and Y. Luo, “Reversible data hiding based on reducing invalid shifting of pixels in histogram shifting,” *Signal Processing*, vol. 163, pp. 238 – 246, 2019. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0165168419301859>
- [94] G. Xuan, Q. Yao, C. Yang, J. Gao, P. Chai, Y. Q. Shi, and Z. Ni, “Lossless data hiding using histogram shifting method based on integer wavelets,” in *Digital Watermarking, 5th International Workshop, IWDW 2006, Jeju Island, Korea, November 8-10, 2006, Proceedings*, ser. Lecture Notes in Computer Science, Y. Shi and B. Jeon, Eds., vol. 4283. Springer, 2006, pp. 323–332. [Online]. Available: https://doi.org/10.1007/11922841_26
- [95] C.-M. Yu, K.-C. Wu, and C.-M. Wang, “A distortion-free data hiding scheme for high dynamic range images,” *Displays*, vol. 32, no. 5, pp. 225–236, dec 2011. [Online]. Available: <https://www-sciencedirect-com.ezproxy.lib.monash.edu.au/science/article/pii/S0141938211000369?via{}%}3Dihub>
- [96] C.-C. Chang, T.-S. Nguyen, and C.-C. Lin, “A new distortion-free data embedding scheme for high-dynamic range images,” *Multimedia Tools and*

Applications, vol. 75, no. 1, pp. 145–163, jan 2016. [Online]. Available: <http://link.springer.com/10.1007/s11042-014-2279-5>

- [97] K. Wong, K. Tanaka, K. Takagi, and Y. Nakajima, “Complete video quality-preserving data hiding,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 19, no. 10, pp. 1499–1512, 2009.
- [98] Zhi-Hui Wang, Chin-Chen Chang, Ting-Yu Lin, and Chia-Chen Lin, “A Novel Distortion-free Data Hiding Scheme for High Dynamic Range Images,” in *2012 Fourth International Conference on Digital Home*. IEEE, nov 2012, pp. 33–38. [Online]. Available: <http://ieeexplore.ieee.org/document/6376380/>
- [99] X. Gao, Z. Pan, E. Gao, and G. Fan, “Reversible data hiding for high dynamic range images using two-dimensional prediction-error histogram of the second time prediction,” *Signal Processing*, vol. 173, p. 107579, 2020. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0165168420301225>
- [100] Y.-T. Lin, C.-M. Wang, W.-S. Chen, F.-P. Lin, and W. Lin, “A Novel Data Hiding Algorithm for High Dynamic Range Images,” *IEEE Transactions on Multimedia*, vol. 19, no. 1, pp. 196–211, jan 2017. [Online]. Available: <http://ieeexplore.ieee.org/document/7558215/>
- [101] J. Fridrich, M. Goljan, and R. Du, “Detecting lsb steganography in color, and gray-scale images,” *IEEE MultiMedia*, vol. 8, no. 4, pp. 22–28, 2001.
- [102] T. Pevny, P. Bas, and J. Fridrich, “Steganalysis by subtractive pixel adjacency matrix,” *IEEE Transactions on Information Forensics and Security*, vol. 5, no. 2, pp. 215–224, 2010.

- [103] N. S. Kulkarni, B. Raman, and I. Gupta, *Multimedia Encryption: A Brief Overview*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 417–449. [Online]. Available: https://doi.org/10.1007/978-3-642-02900-4_16
- [104] J. Yan, T. Chen, and C. Lin, “Encryption in high dynamic range images for RGBE format,” in *2013 Ninth International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, Oct 2013, pp. 493–496.
- [105] P. Puteaux, S. Ong, K. Wong, and W. Puech, “A survey of reversible data hiding in encrypted images – the first 12 years,” *Journal of Visual Communication and Image Representation*, vol. 77, p. 103085, 2021. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S104732032100050X>
- [106] X. Zhang, “Reversible data hiding in encrypted image,” *IEEE Signal Processing Letters*, vol. 18, no. 4, pp. 255–258, April 2011.
- [107] —, “Separable reversible data hiding in encrypted image,” *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 2, pp. 826–832, 2012.
- [108] F. Huang, J. Huang, and Y. Shi, “New framework for reversible data hiding in encrypted domain,” *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 12, pp. 2777–2789, Dec 2016.
- [109] R. Mantiuk, A. Efremov, K. Myszkowski, and H.-P. Seidel, “Backward compatible high dynamic range MPEG video compression,” in *ACM SIGGRAPH 2006 Papers on - SIGGRAPH '06*, vol. 25, no. 3. New York, New York, USA: ACM Press, 2006, p. 713. [Online]. Available: <http://portal.acm.org/citation.cfm?doid=1179352.1141946>
- [110] Ishtiaq Rasool Khan, “Two layer scheme for encoding of high dynamic range images,” in *2008 IEEE International Conference on Acoustics, Speech*

and Signal Processing. IEEE, mar 2008, pp. 1169–1172. [Online]. Available: <http://ieeexplore.ieee.org/document/4517823/>

- [111] M. Iwahashi and H. Kiya, “Efficient lossless bit depth scalable coding for HDR images,” in *Proceedings of The 2012 Asia Pacific Signal and Information Processing Association Annual Summit and Conference*, dec 2012, pp. 1–4.
- [112] —, “Two layer lossless coding of hdr images,” in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, May 2013, pp. 1340–1344.
- [113] A. Skodras, C. Christopoulos, and T. Ebrahimi, “Jpeg2000: The upcoming still image compression standard,” *Pattern Recognition Letters*, vol. 22, no. 12, pp. 1337–1345, 2001, selected Papers from the 11th Portuguese Conference on Pattern Recognition - RECPAD2000. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0167865501000794>
- [114] M. J. Weinberger, G. Seroussi, and G. Sapiro, “The LOCO-I lossless image compression algorithm: principles and standardization into jpeg-ls,” *IEEE Transactions on Image Processing*, vol. 9, no. 8, pp. 1309–1324, Aug 2000.
- [115] E. Reinhard, M. Stark, P. Shirley, and J. Ferwerda, “Photographic tone reproduction for digital images,” *ACM Transactions on Graphics*, vol. 21, no. 3, pp. 267–276, jul 2002. [Online]. Available: <http://portal.acm.org/citation.cfm?doid=566654.566575>
- [116] Y. Mao, G. Chen, and S. Lian, “A novel fast image encryption scheme based on 3d chaotic baker maps,” *International Journal of Bifurcation and Chaos in Applied Sciences and Engineering*, vol. 14, no. 10, pp. 3613–3624, Oct. 2004.
- [117] M. Y. J. TING, “Unified information hiding in high dynamic range image in jpeg xt format,” Aug 2020. [Online]. Avail-

able: https://bridges.monash.edu/articles/thesis/Unified_Information_Hiding_in_High_Dynamic_Range_Image_in_JPEG_XT_format/12894275/1