A Thesis Submitted for the Degree of PhD at the University of Warwick

Inverse Tone Mapping

by

Francesco Banterle, BSc MSc

Submitted to the University of Warwick
for the degree of
Doctor of Philosophy in Engineering

The University of Warwick, School of Engineering

June 4, 2009

Contents

List of Figures

List of Tables

Acknowledgement

Firstly, I would like to thank my supervisor Professor Alan Chalmers for giving me the opportunity for doing a PhD. Alan helped me with precious advices and his encouragement everyday, he was always calm and positive. Even during the bad days, he firstly used to say me ”Don’t panic!”.

I must thank Dr. Patrick Ledda, a friend, that I met in Siggraph 2005, for proposing me as PhD student at Alan’s Laboratory. His advices and help during my PhD were very important for the development of my research on High Dynamic Range Imaging. Another special thank goes to Dr. Prof. Kurt Debattista, a friend, who always gave me good advices, help, and a lift back home. He helped me during my research with fruitful discussions and advices. More important, without him I couldn’t watch the Football World Cup 2006! Last but not least I thank Dr. Alessandro Artusi, a friend, that helped during my research. Both Patrick, Kurt, and Alessandro acted as unofcially secondary advisors during my PhD. Thanks guys for your terrific help!

I have made many friends in Bristol and Warwick over the last past years. I thank: Matt, Gavin, Yusef, Cathy, Veronica, Roger, Anna, Timo, Alexa, Marina, Piotr (ye will), Usama, Tom (Tom), Vibhor (a man of peace), Vedad (Jim), Belma, Jasminka, Jassim, Carlo, Elena, Alena, Remi, Silvester, Gabriela, Keith, Elemedin, and Mike. I had a great time in Warwick, especially at the pubs and other social events held in the Warwickshire. I greatly thank to Alan, Kurt, Alessandro, Paolo, Carlo (hello mate), Gabriela, Elisa, Patrick, Andrea, Keith, and Jassim for reading my thesis and giving me insightful comments and corrections. Thanks mates!!

A special thank goes to the late Usama Mansour, a great man, a contributor of Warwick Visualisation Group, and more important a friend, that tragically died in a friendly football match in October 2007. All the best for his family.

I also thank my friends in Italy for discussions and great moments during my vacations in Italy: Francesco ”Paggio” Pasetto, Alberto ”Bubi” Rodoz, Laura ”Lalla”, Federico Tai, Mauro ”Hawk” Gambini, Sara Migliorini, Roberto Montagna, Matteo ”Caste” Castellarin, and Sebastiano Ridolfi. Moreover, I am grateful to my former supervisors in Italy at Computer Science department at University of Verona, Professor Andrea Fusiello and Professor Roberto Giacobazzi for their help and advices. A special thank to Professor Enrico Gregorio for his help on LaTeX.

Throughout all PhD I had a great help from my family which supported me. This thesis would have never been possible without them. I greatly thank my family, my mother Maria Luisa and my father Renzo, and my brothers Paolo and Piero and his wife Irina. I also thank Elisa, ”ma ufi”, for discussing about Pearls, music, pizza, and Bar Aruuugola.

This work was funded by EPSRC Grant EP/D032148 ”High Dynamic Range for High Fidelity Image Synthesis of Real Scenes” which founded my great stay in United Kingdom.

Declaration

The work in this thesis is original and no portion of the work referred to here has been submitted in support of an application for another degree or qualication of this or any other university or institution of learning.

Part of the work presented in this thesis was previously published by the author in the following papers:

- F. Banterle, P. Ledda, K. Debattista, and A. Chalmers. *Inverse tone mapping*. In GRAPHITE 06: Proceedings of the 4th international conference on Computer graphics and interactive techniques in Australasia and Southeast Asia, pages 349356, New York, NY, USA, 2006. ACM.
- F. Banterle, P. Ledda, K. Debattista, A. Chalmers, and M. Bloj. *A framework for inverse tone mapping*. The Visual Computer, 23(7):467478, 2007.
- F. Banterle, P. Ledda, K. Debattista, and A. Chalmers. *Expanding low dynamic range videos for high dynamic range applications*. In SCCG 08: Proceedings of the 4th Spring Conference on Computer Graphics, pages 349356, New York, NY, USA, 2008. ACM.
- F. Banterle, K. Debattista, P. Ledda, and A. Chalmers. *A gpu-friendly method for high dynamic range texture compression using inverse tone mapping*. In GI 08: Proceedings of graphics interface 2008, pages 4148, Toronto, Ontario, Canada, 2008. Canadian Informa- tion Processing Society.
- F. Banterle, P. Ledda, K. Debattista, A. Artusi, M. Bloj, and A. Chalmers. *A psychophysical evaluation of inverse tone mapping techniques*. To Appear in Computer Graphics Forum, 2009.

In all these publications the author had the role of leader, proposing and developing the main concepts and ideas.

Signed: Date:

Francesco Banterle

Abstract

The introduction of High Dynamic Range Imaging in computer graphics has produced a novelty in Imaging that can be compared to the introduction of colour photography or even more. Light can now be captured, stored, processed, and finally visualised without losing information. Moreover, new applications that can exploit physical values of the light have been introduced such as re-lighting of synthetic/real objects, or enhanced visualisation of scenes. However, these new processing and visualisation techniques cannot be applied to movies and pictures that have been produced by photography and cinematography in more than one hundred years.

This thesis introduces a general framework for expanding legacy content into High Dynamic Range content. The expansion is achieved avoiding artefacts, producing images suitable for visualisation and re-lighting of synthetic/real objects. Moreover, it is presented a methodology based on psychophysical experiments and computational metrics to measure performances of expansion algorithms. Finally, a compression scheme, inspired by the framework, for High Dynamic Range Textures, is proposed and evaluated.

Chapter 1

Introduction

In the last two decades the introduction of High Dynamic Range (HDR) imaging by the computer graphics community has revolutionised the field and other areas such as photography, virtual reality, visual effects, and the video-games industry. Light can now be captured and fully utilised for various applications without the need to linearise the signal and to deal with clamped values. The very dark and bright areas of a scene can be recorded at the same time onto an image or a video, avoiding under-exposed and over-exposed areas. Traditional imaging methods do not use physical values and typically are constrained by limitations in technology that could only handle 8-bit per colour channel per pixel. Such imagery (8-bit or less per colour channel) is known as Low Dynamic Range (LDR) imagery.

The importance of recording light is comparable to the introduction of colour photography. It has changed each stage of the imaging pipeline, see Figure 1.1. In the first stage of the pipeline, capturing, multiple images of the same scene at different exposure levels are recorded, acquiring details from the darkest to the brightest areas. These images are then merged together obtaining an HDR image. There are several algorithms for merging LDR images, for example Debevec and Malik's method [47]. This process can be automated, an example of a commercial automated implementation is the Spheron HDR VR [182] which can capture still spherical images with a dynamic range of $6 \times 10^7 : 1$. However, information could be recorded in one shot using native HDR CCDs, but there is no robust solution at high resolution, without exhibiting low sensor noise at this time.

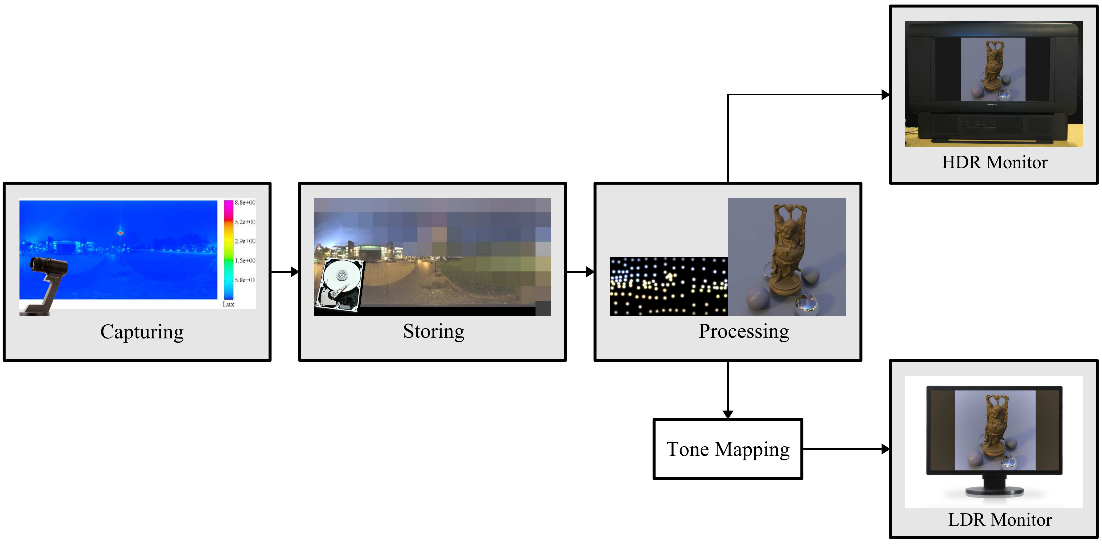


Figure 1.1: *The HDR pipeline in all its stages. Multiple exposure images are captured and combined obtaining an HDR image. Then this image is quantised, compressed, and stored on the hard disk. Further processing can be applied to the image. For example, areas of high luminance can be extracted and used to re-light a synthetic object. Finally, the HDR image or processed ones can be visualised using traditional LDR display technologies or native HDR monitors.*

HDR images/videos occupy four times the amount of memory of uncompressed LDR content. This is because light values are stored using three floating point numbers. This has a huge effect not only on storing and transmitting HDR data, but also in terms of processing performances. Efficient representations of floating point numbers have been designed for HDR imaging, and many classic compression algorithms such as JPEG and MPEG have been extended to handle HDR images and videos.

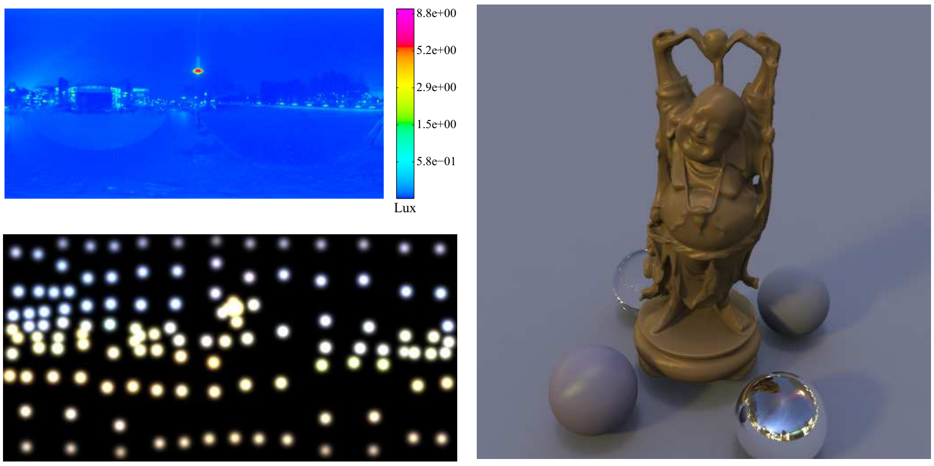


Figure 1.2: *A re-lighting example. On the top left side a spherical HDR image in false colour, and in the bottom light sources extracted from it. On the right side a re-lighted Stanford's Happy Buddha model [73] using those extracted light sources.*

Once HDR content is efficiently captured and stored, it can be utilised for various applications. A very popular application is the re-lighting of synthetic or real objects using HDR images, hence HDR data stores detailed lighting information of an environment. This information can be exploited for detecting light sources and using them for re-lighting objects, see Figure 1.2. Re-lighting is a very important application in many fields such as augmented reality, visual effects, and computer graphics. This is because the appearance of the image is transferred onto the re-lighted objects.


(a) (b)

Figure 1.3: *An example of capturing samples of a BRDF: a) A tone mapped HDR image showing a sample of the BRDF from a Parthenon's block [188]. b) The reconstructed materials in a) from many samples. Images are curtsey of Paul Debevec [188].*

Another important application is to capture samples of the Bi-directional Reflectance Distribution Function (BRDF) which describes how light interacts with a certain material. Then, these samples are used to reconstruct the BRDF, so HDR data needs to be captured for an accurate reconstruction, see Figure 1.3. Moreover, all fields which use LDR imaging can benefit from HDR imaging. For example, disparity calculations in computer vision can be improved in challenging scenes with bright light sources. This is because information in the light sources is not clamped, therefore disparity can be computed for light sources and reflective objects with higher precision than with clamped values.

Once HDR content is created, it needs to be visualised. However, these images/videos do not fit the dynamic range of classic CRT or LCD monitors (or LDR displays/monitors) which is around 200 : 1. Therefore, the HDR content has to be further processed by compressing the dynamic range. This operation is called tone mapping, see Figure 1.4. Recently, monitors that can visualise natively HDR content have been proposed by Seetzeen et al. [179]. The use of these monitors reduces the complexity of further processing of the images/videos.

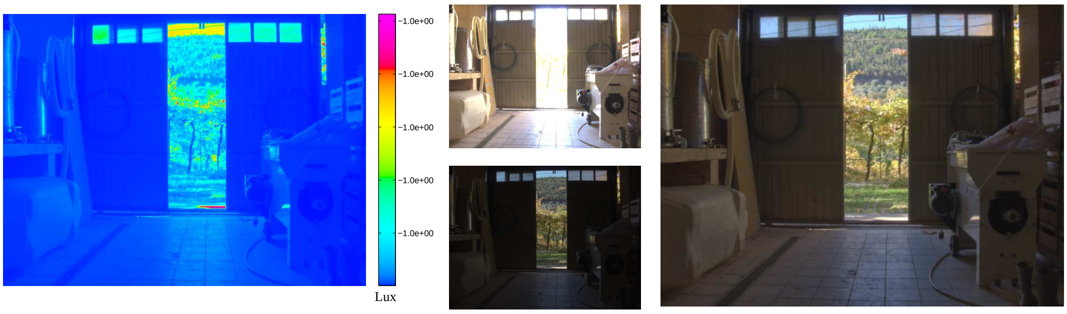


Figure 1.4: *An example of HDR visualisation on a LDR monitor: on the left side an HDR image in false colour. In the centre two slices of the information: in the top an over-exposed image showing details in the dark areas, and in the bottom an under-exposed image showing details in the bright areas. On the right side the image on the left side has been processed to visualise details in bright and dark areas. This processing is called tone mapping.*

1.1 The Need for LDR Content in HDR Applications

The weakest stage in the HDR pipeline is the capturing. At time of writing HDR video-cameras are still an open problem; a robust, noise-free, and high resolution HDR video-camera does not exist. However, there are two main solutions for taking HDR still images. The first one is to use automatic cameras such as the Spheron HDR VR [182], which currently costs around £45,000.

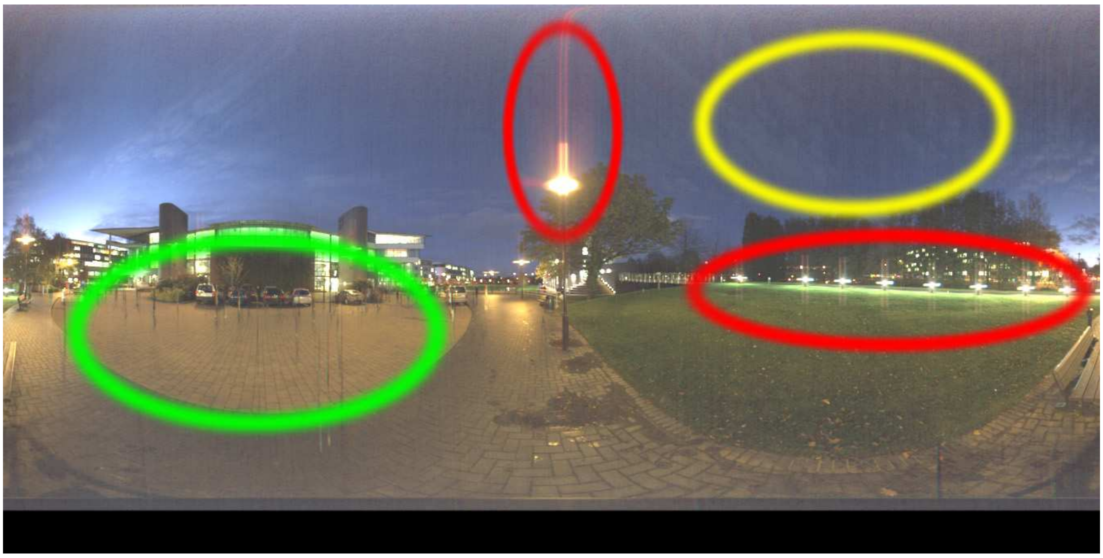
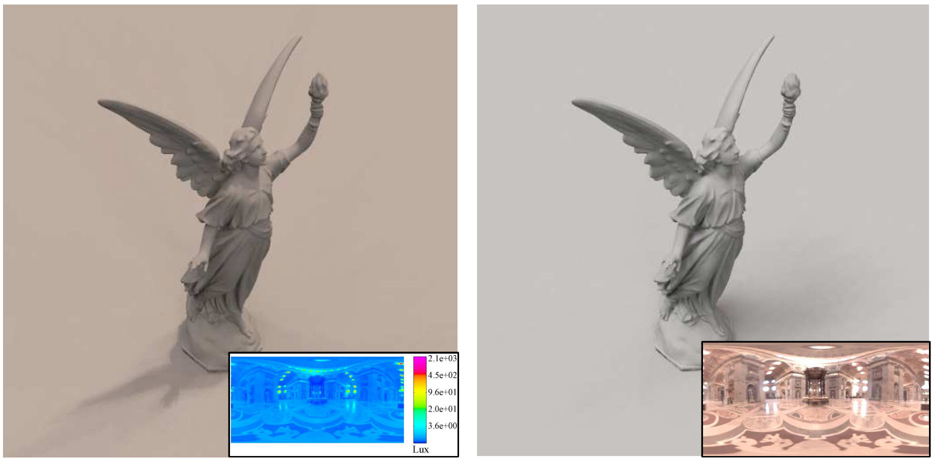


Figure 1.5: *An example of a captured HDR image using Spheron HDR VR. The image has a 12 Megapixels resolution and it was taken in 8 minutes plus 10 minutes for the setup. To minimise moving objects (green circle) shutter speed was set at* $1/30$ *sec. However, the ISO level had to be increased to 800 ISO for compensating the high shutter speed, causing noise (yellow circle). Note that the camera is prone to lens flares (red circle).*

Moreover, with a bulky system and a setup of 10 minutes, it takes on average 8 minutes to capture a scene at 12 Megapixels, see Figure 1.5. The second solution is to use a normal camera. In this case, multiple-exposure images must be taken and then combined into an HDR image, as proposed by Debevec and Malik [47]. This is a long and manual process. A couple of minutes are needed for 6-8 exposures, plus a few minutes for combining the images into an HDR image. Moreover, the camera has to be still during all exposures, therefore a tripod is needed to avoid mis-alignments. However, both these two capturing methods have one common problem: time. Time is essential when capturing a picture, because a scene in the real world is for most of the time dynamic. For example, clouds can obscure the sun in a few seconds, or a car or a person can pass in front of the camera. Moreover, time is a very limited resource in some working fields. For instance, in visual effects the allocated time for capturing data on a cinematographic set is very small and a fast recording method can make a huge difference during post-production.



(a)

(b)

Figure 1.6: *Synthetic objects re-lighting: a) Stanford's Lucy model [73] is re-lighted using an HDR image of St. Peter's Cathedral. b) The same model is re-lighted using an LDR version of the image. Note that colours and hard shadows are lost. The original HDR image is courtesy of Paul Debevec [44].*

Another important aspect of HDR imaging is that there is no link with classic imaging which captures, stores, and processes LDR images or videos. The naïve use of LDR content within HDR applications produces very poor results. For example, a visualised LDR image on an HDR monitor can present quantisation artefacts, see Figure 1.7. Moreover, re-lighting using a LDR image can produce flat and unnatural results, see Figure 1.6. Therefore, LDR images are not suitable for HDR applications. This is a problem, because a huge number of videos and photographs captured in more than one hundred years of photography and cinematography, cannot benefit from HDR applications. For example, an LDR panorama of a place that no longer exists cannot be displayed on HDR monitors or used as source for lighting. Re-lighting and visualisation are important tools for exploring the past and cultural heritage [122, 77]. Moreover, the introduction in the near future of HDR televisions would limit the access of LDR media to consumers.

Figure 1.7: *An example of quantisation errors during visualisation of LDR content on HDR monitors: On the top a LDR image of Venice bay. On the bottom a simulation of the same image on an HDR monitor. Linear stretching produces artefacts in the form of contouring.*

In summary, while capturing HDR images and videos has not yet reached full maturity, LDR imaging has. LDR content at high resolution (more than 10 Megapixels) and at high speed (more than 100 frames per second) can be captured using devices that are cheap and available to consumers. However, LDR content fails to produce good results in HDR applications.

This thesis proposes a solution that fills the following gap: the use of LDR content in HDR applications to skip the expensive and time consuming problem of capturing HDR content. This is achieved by proposing a series of algorithms that expands the dynamic range in LDR images and videos into HDR content on the fly at real-time frame rates. This expansion is followed by a reconstruction part, that generates plausible missing content and reduces quantisation artefacts whenever possible. Furthermore, the approach is validated using psychophysical studies and image metrics. Finally, an application of expansion methods to tone mapped images is exploited to achieve an efficient compression algorithm.

1.2 Contributions

The major contributions of this thesis are:

- An in depth literature review on the field of HDR imaging. All parts of the HDR pipeline are taken into account: image/video capturing, storing, processing, visualisation, relighting, and validation.
- A review of the recent techniques for expanding LDR images/videos. Note that most of these techniques were developed after the publication of the core work presented in this thesis.
- The introduction and formalisation of the inverse tone mapping problem in the computer graphics field. How to generate HDR content from LDR images and videos is the dual of the tone mapping problem.
- A general framework for the expansion of LDR content into HDR that solves the inverse tone mapping problem. This framework can be applied to still images and videos and be accelerated using graphics hardware achieving real-time performances.
- The first study for the evaluation of performances of expansion methods using psychophysical experiments and computational metrics.
- A compression method for HDR textures based on tone mapping and inverse tone mapping. The compression method is evaluated against the state of art in HDR texture compression.

1.3 Thesis Structure

The thesis is structured in seven chapters as follows:

- **Chapter 2: Background** introduces an overview of the High Dynamic Range Imaging pipeline: how to create, store, process, and visualise HDR images and videos. Moreover, a brief introduction on light quantities, human visual system, and colour spaces is provided.
- **Chapter 3: Inverse Tone Mapping Operators** is an in depth state of art on expansion methods for legacy content and compression schemes.
- **Chapter 4: Methodology** introduces the methodology which was used for the developing of the new algorithm for expanding LDR content into HDR content.
- **Chapter 5: An Inverse Tone Mapping Operator** describes the framework of the thesis, proposing a version for still images, videos, and an optimised implementation that exploits the power of modern graphics hardware.
- **Chapter 6: Validation of Inverse Tone Mapping Operators** presents two studies for the evaluation of performances of iTMOs. The first one is a pyschophysical study that employed paired comparisons. The second study is based on a perceptual metric.
- **Chapter 7: Inverse Tone Mapping Application: HDR Content Compression** is an application of inverse tone mapping to compress still images. While encoding is realised through tone mapping, decoding is achieved using inverse tone mapping. Further compression is achieved using a standard compression scheme, which is applied to tone mapped images.
- **Chapter 8: Conclusions and Future Work** summaries the thesis, showing the main results and limitations. Future work in the field of inverse tone mapping is outlined with an emphasis on how to exploit temporal coherence to improve detail in expanded regions.

Chapter 2

High Dynamic Range Imaging

HDR imaging is a revolution in the field of imaging, because it has introduced the use of physical-real values of light. This chapter introduces how to capture, encode, visualise and use HDR content. The chapter is structured into six sections:

- Section 2.1.1. **Light, Human Vision, and Colour Space**: This section introduces basic concepts of light, human vision, and colour spaces.
- Section 2.2. **The Generation of HDR Images and Videos**: This section describes the main techniques for generating HDR images and videos.
- Section 2.3. **Encoding HDR Images and Videos**: HDR content needs more memory to be stored than LDR content, because more information is needed to represent the full range of real world lighting. This raises the problem of efficiently storing images and videos. This section introduces the main encoding and compression schemes for HDR images and videos.
- Section 2.4. **Tone Mapping**: HDR content exceeds the dynamic range of current display technology such as CRT and LCD. Therefore, the visualisation of HDR images and videos on classic CRT and LCD displays is achieved by compressing intensities in the amplitude of the signal. This process is called tone mapping.
- Section 2.5. **Native Visualisation of HDR Content**: To provide a native visualisation of HDR content, researchers have developed HDR display technologies such as the HDR

viewer and HDR displays. This section introduces these technologies that have been proposed in the last few years.

- Section 2.6. **Evaluation of Tone Mapping Techniques**: The large number of tone mapping techniques introduced the need to determine which operator performs better than others for certain tasks, scenarios, and situations. This section introduces evaluation studies that have been proposed to measure the performances of tone mapping algorithms.
- Section 2.7. **Image Based Lighting**: One of the main applications of HDR content is image based lighting (IBL), which allows the realistic re-lighting of virtual and real objects. An overview on IBL is presented with emphasis on how to solve it.

2.1 Light, Human Vision, and Colour Spaces

This section introduces basic concepts of visible light and units for measuring it, the human visual system (HVS) focusing on the eye, and colour spaces. These concepts are very important in HDR imaging as it deals with physical-real values of light, from very dark values (i.e. 10^{-3} cd/m^2) to very bright ones (i.e. 10^6 cd/m^2). Moreover, the perception of a scene by the HVS depends greatly on the lighting conditions.

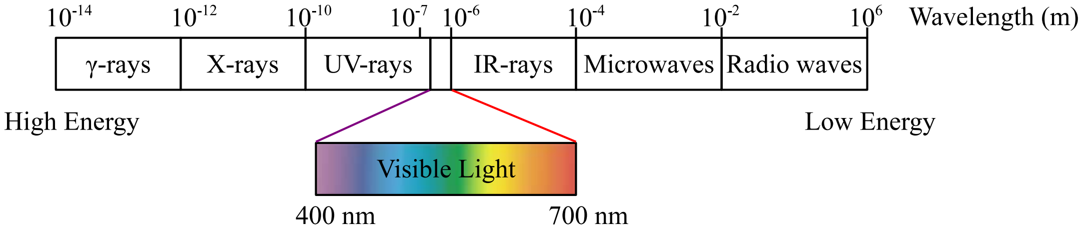


Figure 2.1: *The elctromagnetic spectrum. The visible light has a very limited spectrum between 400 nm and 700 nm.*

2.1.1 Light

Visible light is a form of radiant energy which has a dual nature. It is at the same time an electromagnetic wave as shown by Maxwell and a particle as shown by Einstein [65]. Light travels in the space interacting with materials where it can be absorbed, refracted, reflected, and transmitted, see Figure 2.2. While the light is travelling, it can reach human eyes, stimulating them and producing visual sensations depending on the wavelength, see Figure 2.1.

Radiometry and Photometry define how to measure light and its units over time, space or angle. While the former measures physical units, the latter takes into account the human eye, where spectral values are weighted by the spectral response of the eye ($\bar{y}$ curve, see Figure 2.4). Radiometry and Photometry units were standardised by the Commission Internationale de l'Eclairage (CIE) [34]. The main radiometric units are:

- **Radiant Energy** (Ω_e): is the basic unit for light, it is measured in joules (J).
- **Radiant Power** ($P_e = \frac{\Omega_e}{dt}$): is the amount of energy that flows per unit of time (Js^{-1}=W).
- **Radiant Intensity** ($I_e = \frac{dP_e}{d\omega}$): is the amount of Radiant Power per unit of direction (Wsr^{-1}).
- **Irradiance** ($E_e = \frac{dP_e}{dA_e}$): is the amount of Radiant Power per unit of area from all direction of the hemisphere at a point (Wm^{-2}).
- **Radiance** ($L_e = \frac{d^2 P_e}{dA_e \cos\theta d\omega}$): is the amount of Radiant Power arriving/leaving at a point in a particular direction ($\mathrm{Wm}^{-2}\mathrm{sr}^{-1}$).

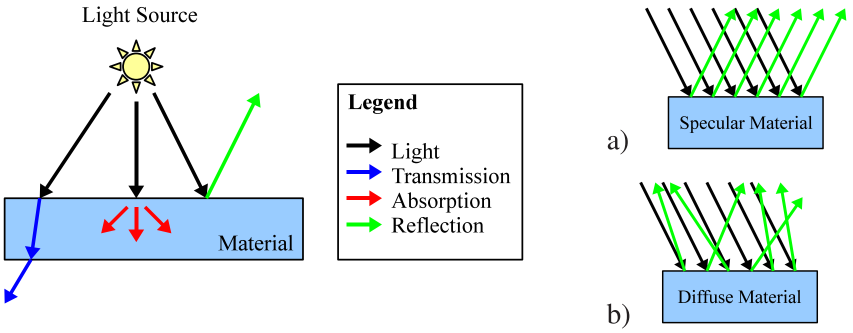


Figure 2.2: *On the left side the three main light interactions: transmission, absorption, and reflection. In transmission, light travels through the material changing its direction according to the physical properties of the medium. In absorption, the light is taken up by the material that was hit and it is converted into thermal energy. In reflections, light bounces from the material in a different direction due the material's properties. There are two main kind of reflections: specular and diffuse. On the right side: a) Specular reflections; a ray is reflected in a particular direction. b) Diffuse reflections; a ray is reflected in a random direction.*

The main photometric units are:

- **Luminous Power** (P_v): is the weighted Radiant Power, it is measured in lumens (lm) a derived unit from candela (lm= cd sr).
- **Luminous Energy** (Q_v): is the analogous of the Radiant Energy (lm s).
- **Luminous Intensity** (I_v): is the Luminous Power per direction, it is measured in candela (cd or lm sr^{-1}).
- **Illuminance** (E_v): is the analogous of the Irradiance (lm m^{-2}).
- **Luminance** (L_v): is the weighted Radiance (lm m^{-2}sr^{-1} or cd m^{-2}).

A measure of the relative luminance of the scene can be useful, hence it can help to understand some properties of the scene such as the presence of diffuse or specular surfaces, lighting condition, etc. For example, specular surfaces reflect light sources even if they are not visible directly in the scene, increasing the relative luminance. This relative measure is called *Contrast*. Contrast is formally a relationship between the darkest and the brightest value in a scene, and it can be calculated in different ways. The main contrast relationships are Weber Contrast, C_{W}, Michelson Contrast , C_{M}, and Ratio Contrast, C_{R}. These are defined as:

$$C_{\mathrm{W}} = \frac{L_{\mathrm{Max}} - L_{\mathrm{Min}}}{L_{\mathrm{Min}}} \qquad C_{\mathrm{M}} = \frac{L_{\mathrm{Max}} - L_{\mathrm{Min}}}{L_{\mathrm{Max}} + L_{\mathrm{Min}}} \qquad C_{\mathrm{R}} = \frac{L_{\mathrm{Max}}}{L_{\mathrm{Min}}} \tag{2.1}$$

where L_{Min} and L_{Max} are respectively the minimum and maximum luminance values of the scene. In this thesis C_{R} is used as contrast definition.

2.1.2 An Introduction to the Human Eye

The eye is an organ which gathers light onto photoreceptors which convert light into electrical signals, see Figure 2.3. These are transmitted through the optical nerve to the visual cortex, an area of the brain that processes these signals producing the visual image. This full system, which is responsible for vision, is referred to as HVS.

Light, which enters in the eye, firstly passes through the Cornea, a transparent membrane. Then it enters into the Pupil, an aperture which is modified by the Iris, a muscular diaphragm. Subsequently, light is refracted by the Lens, and hits photoreceptors in the Retina. Note that inside the eye there are two liquids, vitreous and aqueous humours. The former fills the eye keeping its shape and the Retina against the inner wall. The latter is between the Cornea and the Lens and maintains the intraocular pressure.

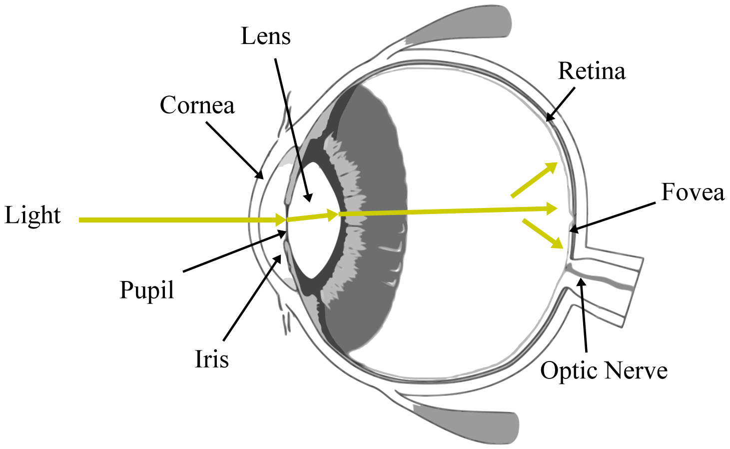


Figure 2.3: *The human eye. A modified image from Mather [129].*

There are two types of photoreceptors, cones and rods. The cones, number around 6 million, are located in the Fovea. They are sensitive at luminance levels between 10^{-2} cd/m^2 and around 10^6 cd/m^2 (Photopic vision or daylight vision), and responsible for the perception of high frequency pattern, fast motion, and colours. Furthermore, colour vision is due to three types of cones: short wavelength cones, sensitive to wavelengths around 435nm, middle wavelength cones, sensitive around 530nm, and long wavelength cones, sensitive around 580nm. The rods, number around 90 million, are sensitive at luminance levels between 10^{-2} cd/m^2 and 10^{-6} cd/m^2 (Scotopic vision or night vision). Moreover, there is only one type of rod, which does not mediate colours limiting the ability to distinguish colours. They are located around the Fovea, but absent in it. This is why high frequency patterns cannot be distinguished at low light conditions. Note that an adaptation time is needed for passing from Photopic to Scotopic vision and viceversa, for more details see [129]. The rods and cones compress the original signal reducing the dynamic range of incoming light. This compression has a sigmoid which can be fitted in the following model:

$$\frac{R}{R_{\text{Max}}} = \frac{I^n}{I^n + \sigma^n} \tag{2.2}$$

where R is the photoreceptor response, R_{Max} is the maximum response, I is the light intensity.

sigma and n are respectively the semi-saturation constant and the sensitivity control exponent, which are different for cones and rods [170, 129].

2.1.3 Colour Spaces

A colour space is a mathematical description for representing colours, typically as three components such as in the case of RGB and XYZ which are called primary colours. A colour space is usually defined taking into account human perception and the capabilities to display colours of a device which can be a LCD monitor, a CRT monitor, paper, etc.

One of the first proposed colour spaces was CIE 1931 XYZ colour space, which is based on the response of short (S), middle (M) and long (L) wavelength rods' responses. The concept is that a colour sensation can be described as an additive model based on the amount of three primary colours (S, M, and L). XYZ is formally defined as the projection of a spectral power distribution I, into the responses of rods or colour-matching functions, $\bar{x}$, $\bar{y}$, and $\bar{z}$:

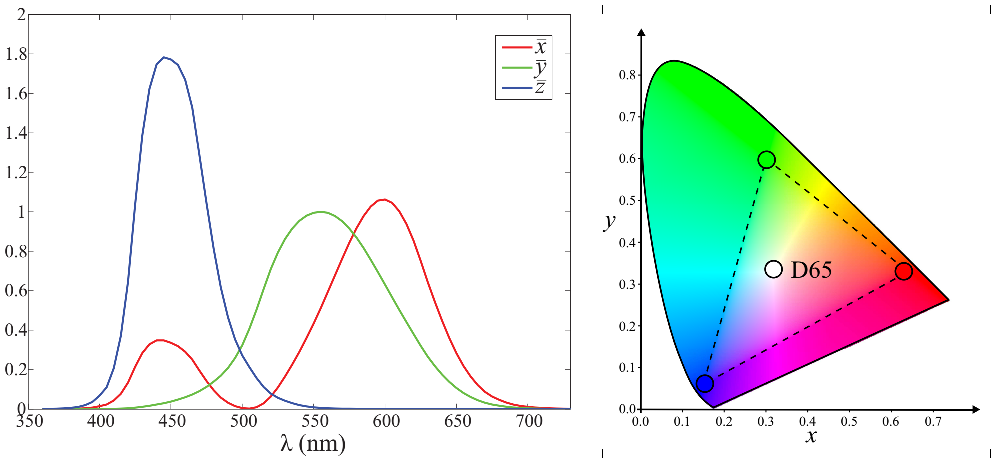
$$X = \int_{380}^{830} I(\lambda)\bar{x}(\lambda)d\lambda \quad Y = \int_{380}^{830} I(\lambda)\bar{y}(\lambda)d\lambda \quad Z = \int_{380}^{830} I(\lambda)\bar{z}(\lambda)d\lambda \tag{2.3}$$

$\bar{x}$, $\bar{y}$, and $\bar{z}$ are plotted in Figure 2.4. Note that XYZ colour space was designed in such a way that the Y component measures the luminance of the colour. The information relative for the hue and colourfulness of the colour or chromaticity is derived from XYZ values as:

$$x = \frac{X}{X+Y+Z} \quad y = \frac{Y}{X+Y+Z} \tag{2.4}$$

These values can be plotted, producing a representation of the colours that HVS can perceive which is called gamut, see Figure 2.4.b.

A popular colour space for CRT and LCD monitors is sRGB [185] and it was adopted in this thesis. This colour space defines as primaries colours the red (R), green (G), and blue (B). Moreover, each colour in sRGB is a linear additive combination of values in $[0,1]$ of the three primaries. Therefore, not all colours can be represented but only the one inside the triangle generated by the three primaries, see Figure 2.4.b .



(a) (b)

Figure 2.4: *The CIE XYZ colour space: a) The CIE 1931 2-degree /XYZ colour matching functions. b) The CIE xy chromaticity diagram showing all colours that HVS can perceive. Note that the triangle is the space of colour that can be represented in sRGB, where the three circles represent the three primaries.*

Between XYZ and RGB colour space there exists a linear relationship, therefore RGB colours can be converted into XYZ ones using the following conversion matrix **M**:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \mathbf{M} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \qquad \mathbf{M} = \begin{bmatrix} 0.412 & 0.358 & 0.181 \\ 0.213 & 0.715 & 0.072 \\ 0.019 & 0.119 & 0.950 \end{bmatrix} \tag{2.5}$$

Furthermore, sRGB presents a non-linear transformation for each R, G, and B channel to linearise the signal when displayed on LCD and CRT monitors. This is because there is a non-linear relationship between the output intensity generated by the displaying device and the input voltage. This relationship is generally approximated with a power function with value $\gamma = 2.2$. Therefore, the linearisation is achieved by applying the inverse value:

$$\begin{bmatrix} R_v \\ G_v \\ B_v \end{bmatrix} = \begin{bmatrix} R \\ G \\ B \end{bmatrix}^{\frac{1}{\gamma}} \tag{2.6}$$

where R_v, G_v, B_v are respectively red, green, and blue channels ready for the visualisation. This process is called gamma correction.

In HDR imaging RGB colour space is very popular. However, many computations are calculated in the luminance channel Y from XYZ which is usually referred to as L. Moreover, common statistics from this channel are employed such as the maximum value, L_{Max}, the minimum one L_{Min}, and the mean value. This can be computed as arithmetic average, L_{Avg}, or geometric one, L_{H}:

$$L_{\text{Avg}} = \frac{1}{N}\sum_{i=1}^{N} L(\mathbf{x}_i) \qquad L_{\text{H}} = \exp\left(\frac{1}{N}\sum_{i=1}^{N} \log\bigl(L(\mathbf{x}_i)+\varepsilon\bigr)\right) \tag{2.7}$$

where $\mathbf{x}_i$ are the coordinates of the i-th pixel, and $\varepsilon > 0$ is a small constant for avoiding singularities. Note that in HDR imaging subscript $_{\text{w}}$ and $_{\text{d}}$ respectively refer to HDR and LDR values. The main symbols used in HDR image processing are shown in Table 2.1 for the luminance channel L.

Symbol	Description
L_{w}	HDR luminance value
L_{d}	LDR luminance value
L_{H}	Harmonic mean luminance value
L_{Avg}	Arithmetic mean luminance value
L_{Max}	Maximum luminance value
L_{Min}	Minimum luminance value

Table 2.1: *The main symbols used for the luminance channel.*

2.2 The Generation of HDR Images

HDR content can be generated using computer graphics, photography, or an augmentation of both. Synthesising images with computer graphics takes a big effort in terms of specification of the scene. This is because each component needs to be modelled such as geometry, materials, light sources, and how light is transported. On the other hand, photography is an easier process than synthesis, because a scene can be found and captured through a simple process. However, HDR content needs time consuming techniques and expensive equipment in order to be captured.

2.2.1 Synthesising HDR Images and Videos from the Virtual World

The synthesis of images and videos in computer graphics is called rendering, where a formal description of the scene is needed to synthesise the content. Two main algorithms are usually employed for rendering: ray tracing and rasterisation, see Figure 2.6.

Ray Tracing

Introduced by Whitted [219], ray tracing is a very elegant algorithm which shoots for each pixel in the screen a ray according to the direction of the camera and its lens, see Figure 2.5. This ray traverses the scene until it intersects or hits an object, which is equivalent to solving a non linear system of equations. At the hit point, the lighting is evaluated according to the material properties or BRDF, and light sources. To determine if a point is in shadow or in light, a ray is shot from the point toward the direction of the light source. Then, according to the BRDF more rays, called secondary rays, are shot in the scene to simulate reflections, refractions, inter-reflections; in general the light transport [91]. This process stops when it converges to a stable value, or when a threshold set by the user is reached.

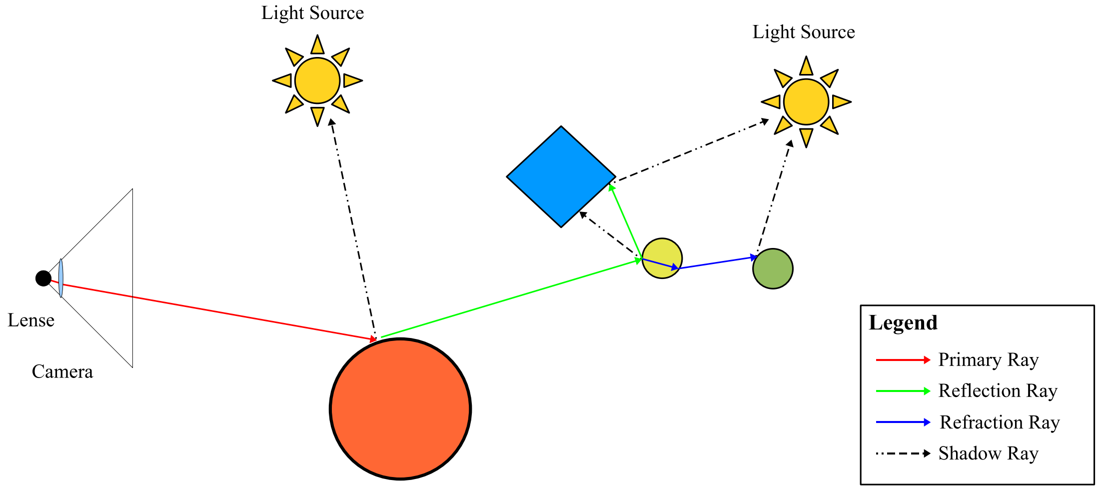


Figure 2.5: *Ray tracing: for each pixel in the image a primary ray is shot through the camera in the scene. As soon as it hits a primitive, the lighting for the hit point is evaluated. This is achieved by shooting more rays. For example, a ray towards the light is shot in the evaluation of lighting. A similar process is repeated for reflection, refractions and inter-reflections.*

The main advantage of ray tracing is the ability to simulate physically based effects such as shadows, reflections, refractions, inter-reflections etc. On the other hand, the main disadvantage is the difficulty to accelerate it on current hardware. The main problem is due to data structures, which are slow to update in the case of fully deformable geometry [203]. Moreover, the secondary rays are not coherent in their traversal of the scene, a problem that can cause cache misses and low performance rates [203].


(a) (b)

Figure 2.6: *An example of state of art of rendering quality for ray tracing and rasterisation: a) A raytraced image by by Piero Banterle using Maxwell Render from NextLimit Technologies [144]. b) A screenshot from the game Crysis by Crytek GmbH [38].*

Rasterisation

Rasterisation is a different approach to solve the rendering problem compared to ray tracing. The main concept is to project each primitive that composes the scene on the screen (frame buffer) and discretise it into fragments. This operation is called scan conversion, for more detail see [66]. When a primitive is projected and discretised, visibility has to be solved to have a correct visualisation and to avoid overlaps between objects. For this task the Z-buffer [28] is generally used. The Z-buffer is an image of the same size of the frame buffer that stores depth values of previous solved fragments. For each fragment at a position $\mathbf{x}$, its depth value, $F(\mathbf{x})_z$, is tested against the stored one in the Z-buffer, $Z(\mathbf{x})_z$. If $F(\mathbf{x})_z < Z(\mathbf{x})_z$, the new fragment is written in the frame buffer, and $F(\mathbf{x})_z$ in the Z-buffer. After the depth test, lighting is evaluated for all fragments. However, shadows, reflections, refractions, and inter-reflections are not possible to simulate natively, because rays cannot be shot. The solution is to render to a texture the scene from different positions to emulate these effects. For example, shadows are emulated by calculating a Z-buffer from the light source position, and applying a depth test during shading to determine if the point is in shadow or not [221].

The main advantage is the support by current graphics hardware, which allows high performances in terms of drawn primitives. Hence, the algorithm is easy to parallelise: fragments are coherent and independent, and data structures are easy to update. Finally, the whole process is easy to organise into a pipeline. Nevertheless, the emulation of physically based light transport effects (i.e. shadows, reflections/refractions, etc.) is not precise, not accurate, and biased in many cases.

2.2.2 Capturing HDR Images and Videos from the Real World

Nowadays, available consumer cameras are limited in that they can only capture 8-bit images or 12-bit in RAW format, which do not cover the full dynamic range of irradiance values in most environments in the real world. The only way to capture HDR is to take multiple exposure images of the same scene for capturing details from the darkest to the brightest areas as proposed by Mann and Picard [123], see Figure 2.7 for an example. If the camera has a linear response, the radiance values stored in each exposure for each colour channel can be combined to recover the irradiance, E, as:

$$E_{\mathrm{k}}(\mathbf{x}) = \frac{\sum_{i=1}^{N_e} \frac{1}{\Delta t_i} w(I_{i,\mathrm{k}}(\mathbf{x})) I_{i,\mathrm{k}}(\mathbf{x})}{\sum_{i=1}^{N} w(I_{i,\mathrm{k}}(\mathbf{x}))} \tag{2.8}$$

where I_i is the image at the i-th exposure, k is the index of the colour channel for I_i, Δt_i is the exposure time for I_i, N_e is the number of images at different exposures, and $w(I_{i,\mathrm{k}}(\mathbf{x}))$ is a weight function that chooses pixel values to remove outliers. For example, high values can be preferred to have less noise that affects in low values. On the other hand, high values can be saturated, so middle values can be more reliable. An example of a recovered irradiance map using Equation 2.8 can be seen in Figure 2.7.f.

The problem with film and digital cameras is that they do not have a linear response, but a more general function f, called camera response function (CRF). This is due to the fact that the dynamic range of real world does not fit the medium, so as much as possible data is fitted into 8-bit or into film using f. Mann and Picard [123] proposed a simple method for calculating f, which consists of fitting the values of pixels at different exposure to a fixed CRF, $f(x) = ax^{\gamma} + b$. This parametric f is very limited and does not support most real CRFs.

Debevec and Malik [47] proposed a simple method for recovering a CRF. The value of a pixel in an image is given by the application of a CRF to the irradiance scaled by the exposure time:

$$I_{\mathrm{k}}(\mathbf{x}) = f(E_{\mathrm{k}}(\mathbf{x})\Delta t_i) \tag{2.9}$$

(a) (b) (c) (d) (e) (f)

Figure 2.7: *An example of HDR capturing of the Stanford Memorial Church. Images taken with different shutter speeds: a)* $\frac{1}{250}$ *sec. b)* $\frac{1}{30}$ *sec. c)* $\frac{1}{4}$ *sec. d) 2 sec. and e) 8 sec. The final HDR image obtained by combining a), b), c), d), e): f) A rendering in false colour of the luminance channel. The original HDR image is courtesy of Paul Debevec [47]*

If terms are re-arranged, and a logarithm is applied to both side, the results is:

$$\log(f^{-1}(I_{\mathrm{k}}(\mathbf{x}))) = \log E_{i,\mathrm{k}}(\mathbf{x}) + \log \Delta t_i \tag{2.10}$$

Assuming that f is a smooth and monotonically increasing function, f and E can be calculated by minimising the least square error derived from Equation 2.10 using pixels from images at different exposure:

$$\mathscr{O} = \sum_{i=1}^{N_e} sum_{j=1}^{M} \Big(w\big(I_{i,\mathrm{k}}(\mathbf{x}_j)\big) \big[g(I_{i,\mathrm{k}}(\mathbf{x}_j)) - \log E_{\mathrm{k}}(\mathbf{x}_j) - \log \Delta t_i \big] \Big)^2 + \lambda \sum_{x=T_{\min}+1}^{T_{\max}-1} (w(x) g''(x))^2 \tag{2.11}$$

where $g = f^{-1}$ is the inverse of the CRF, M is the number of pixel used in the minimisation, and $T_{\max}$ and $T_{\min}$ are respectively the maximum and minimum integer values in all images I_i. The second part of Equation 2.11 is a smoothing term for removing noise, where function w is defined as:

$$w(x) = \begin{cases} x - T_{\min} & \text{if } x \leq \frac{1}{2}(T_{\max} + T_{\min}) \\ T_{\max} - x & \text{if } x > \frac{1}{2}(T_{\max} + T_{\min}) \end{cases} \tag{2.12}$$

Note that minimisation is performed only on a subset of M pixel, because it is computationally expensive to evaluate for all pixels. This subset is calculated using samples from each region of the image. Mitsunaga and Nayar [137] improved Debevec and Malik's algorithm with a more robust method based on a polynomial representation of f.

The multiple exposure method assumes that images are perfected aligned, there are no moving objects, and CCD noise is not a problem. These are very rare conditions when real world images are captured. These problems can be avoided by adapting classic alignment, ghost and noise removing techniques from image processing and computer vision, for details see [170].

HDR videos can be captured using still images, with techniques such as stop-motion or time-lapse. Nevertheless, these techniques work well in controlled scenes. To solve the video capturing problem Kang et al. [92] extended multiple exposure images methods to videos. The concept is to have a programmed video-camera that temporally varies the shutter speed at each frame. The final video is generated aligning and warping different frames, for combining 2 frames into a HDR one. However, the frame rate of this method is low, around 15 fps, and the scene has to contain low speed moving objects otherwise artifacts will appear. Therefore, the method is not suitable for real world situations. Nayar and Branzoi [141] developed an adaptive dynamic range camera, where a controllable liquid crystal light modulator is placed in front of the camera. This modulator adapts the exposure of each pixel on the image detector allowing to capture scenes with a very large dynamic range.

In the commercial field, a few companies provide HDR cameras based on automatic multiple exposure capturing. The two main cameras are Spheron HDR VR camera [182] by SpheronVR GmbH and Panoscan MK-3 [150] by Panoscan Ltd, which are both full 360 degree panoramic cameras at high resolution. The two cameras take full HDR images. For example, Spheron HDR VR can capture 26 f-stops of dynamic range at 50 Megapixels resolution. The development of these particular cameras was due to the necessity of quickly capturing HDR images for IBL, see Section 2.7, which is extensively used in visual effects, computer graphics, automotive and more in general for product advertising.

The alternative to multiple exposure techniques is to use CCD sensors which can natively capture HDR values. In recent years, CCDs that record into 10/12-bit channels in the logarithmic domain have been introduced by many companies such as Cypress Semiconductor [39], Omron [147], PTGrey [164], Neuricam [143], etc. The main problem with these sensors is that they are at low resolution (640×480), and very noisy. Therefore, their applications are oriented to automotive, security, and automatisation in factories.

In the cinema industry a few companies have proposed high quality solutions such as Viper camera [189] by Thomson GV, Red One camera [167] by RED Company, and the Phantom HD camera [202] by Vision Research. All these video-cameras present high frame rates, low noise, full HD ($1,920 \times 1,080$) resolution, and a good dynamic range, 10/12-bit per channel in the logarithmic/linear domain. However, they are extremely expensive (sometimes available only for renting) and they do not fit the full dynamic range of the HVS.

2.3 Encoding HDR Images and Videos

Once HDR images/videos are captured from the real world, or they are synthesised using computer graphics, there is the need to store, distribute, and process these images. An uncompressed HDR pixel is represented using three single precision floating point numbers [82], assuming three bands as for RGB colours. This means that a pixel uses 12 bytes of memory, and an HD image $1,920 \times 1,080$ occupies 23 Mb, an extremely large amount of space. Researchers have been working on compression methods for HDR content to address the high memory demands. At the beginning, only compact representations of floating point numbers were proposed, see Section 2.3.1. In the last few years, researchers have focused their efforts into more complex compression schemes, see Section 2.3.2.

2.3.1 HDR Pixels Representation in Floating Point Formats

HDR values are usually stored using single precision floating point numbers. Integer numbers, which are extensively used in LDR imaging, are not practical for storing HDR values. For example, a 32-bit unsigned integer can represent values in the range $[0, 2^{32} - 1]$, which seems to be enough for most HDR content. However, when a simple image processing between two or more images is carried out, for example an addition or multiplication, precision can be easily lost and overflow can occur. These are classic problems that argue in favour of floating point numbers rather than integer ones for real world values [82].

The main problem of single precision floating point numbers is that an image occupies 96 bits per pixel (bpp), which is four times the amount of space needed for an uncompressed LDR image. Ward [209] proposed the first solution to this problem, RGBE, which was meant for storing HDR values generated by the Radiance rendering system [216]. The concept is to have a shared exponent between the three colours, assuming that it does not vary much between them. The encoding of the format is defined as:

$$E = \left\lceil \log_2\left(max(R_\mathrm{w}, G_\mathrm{w}, B_\mathrm{w})\right) + 128 \right\rceil$$
$$R_\mathrm{m} = \left\lfloor \frac{256R_\mathrm{w}}{2^{E-128}} \right\rfloor \quad G_\mathrm{m} = \left\lfloor \frac{256G_\mathrm{w}}{2^{E-128}} \right\rfloor \quad B_\mathrm{m} = \left\lfloor \frac{256B_\mathrm{w}}{2^{E-128}} \right\rfloor \tag{2.13}$$

and the decoding as:

$$R_\mathrm{w} = \frac{R_\mathrm{m}+0.5}{256}2^{E-128} \quad G_\mathrm{w} = \frac{G_\mathrm{m}+0.5}{256}2^{E-128} \quad B_\mathrm{w} = \frac{B_\mathrm{m}+0.5}{256}2^{E-128} \tag{2.14}$$

Mantissas of the red, R_m, green, G_m, and blue, B_m, channels and the exponent, E, are then stored in an unsigned char (8-bit), achieving a final format of 32 bpp. The RGBE encoding covers 76 orders of magnitude, but the encoding does not support the full gamut of colours and negative values. To solve these problems, an image is converted to the XYZ colour space before encoding. This case is referred to as the XYZE format. Recently, the RGBE format has been implemented in graphics hardware on the NVIDIA G80 series [95], allowing very fast encoding/decoding for real-time applications.

In addition, Ward proposed a 24 and 32 bpp perceptually based format, the LogLuv format [106]. The concept is to assign more bits to luminance in the logarithmic domain than to colours in the linear domain. Firstly, an image is converted to the colour space which separates luminance (Y of XYZ) and chromaticity (CIE 1976 Uniform chromaticity (u, v)). The 32 bpp format assigns 15 bits to luminance and 16 bits to chromaticity, and it is defined as:

$$L_e = \left\lfloor 256(\log_2 Y_\mathrm{w} + 64) \right\rfloor \qquad u_e = \left\lfloor 410u' \right\rfloor \quad v_e = \left\lfloor 410v' \right\rfloor \tag{2.15}$$

and the decoding as:

$$Y = 2^{(L_e+0.5)/256-64} \qquad u' = \left\lfloor \frac{1}{410}(u_e) \right\rfloor \quad v' = \left\lfloor \frac{1}{410}(v_e) \right\rfloor \tag{2.16}$$

Note that the 24 bpp format changes slightly in the constants of Equation 2.15 and Equation 2.16 according with the distribution of bits: 10 bits are allocated to luminance and 14 bits to chromaticity. While the 32 bpp format covers 38 orders of magnitude, the 24 bpp format achieves only 4.8 orders. The main advantage, over RGBE/XYZE is that the format is already in a separate colour space. Therefore, values are ready to be used for applications such as tone mapping, see Section 2.4, colour manipulation, etc.

Another important format is the half floating point format which is part of the specification of the OpenEXR file format [86]. In this representation each colour is encoded using an half precision number, which is a 16-bit implementation of the IEEE 754 standard [82], and it is defined as:

$$H = \begin{cases} 0 & \text{if } \big(M = 0 \wedge E = 0\big) \\ (-1)^S 2^{E-15} + \frac{M}{1024} & \text{if } E = 0 \\ (-1)^S 2^{E-15}\left(1 + \frac{M}{1024}\right) & \text{if } 1 \leq E \leq 30 \\ (-1)^S \infty & \text{if } \big(E = 31 \wedge M = 0\big) \\ \text{NaN} & \text{if } \big(E = 31 \wedge M > 0\big) \end{cases} \tag{2.17}$$

where S is the sign, 1 bit, M is the mantissa, 10 bits, and E is the exponent, 5 bits. Therefore, the final format is a 48 bpp, covering around 10.7 orders of magnitude. The main advantage despite the size, is that this format is implemented in graphics hardware by main vendors allowing real-time applications to use HDR images. Also, this format is considered as the de facto standard in the movie industry as the new digital negative [46].

The cinema industry proposed several medium dynamic range formats, which have the purpose of covering classic film range between 2-4 orders of magnitude, but they are not suitable for HDR images/videos, see for example the Pixar Log encoding [46].

2.3.2 HDR Image and Texture Compression

While a good floating point format can achieve down to 32/24 bpp (RGBE and LogLuv), this is not enough for distributing HDR images/videos or storing a large database of images, etc. Therefore, researchers have been working on modifying current image compression standards or proposing new methods for reducing the size of HDR images and videos. This section presents a review of the main algorithms for the compression of HDR images/videos. The focus of this review is on the compression of HDR textures, which are images used in computer graphics for increasing details of materials.

Note that some compression techniques are described in Section 3.2. These have been moved to be more coherent with the topic of Inverse Tone Mapping. These schemes are: backward compatible HDR-MPEG [125], JPEG-HDR [213, 214], HDR-JPEG 2000 [223], compression through sub-bands [115], encoding using a model of human cones [75], and two layer encoding [146].

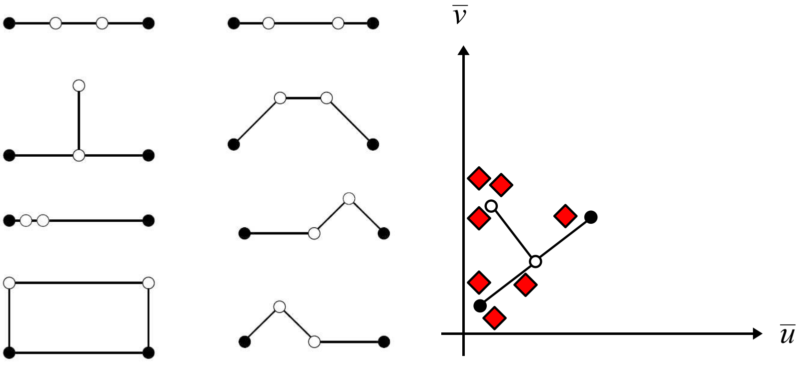
HDR Textures Compression using Geometry Shapes

A block truncation compression (BTC) scheme [79] for HDR textures was proposed by Munkberg et al. [139]. This scheme compresses 48 bpp HDR textures into 8 bpp leveraging on logarithmic encoding of luminance and geometry shape fitting for the chrominance channel.

The first step in the decoding scheme is to transform data in a colour space where luminance and chrominance are separated to achieve a better compression. For this task, the authors defined a colour space, $\overline{Y}\overline{u}\overline{v}$, which is defined as:

$$\begin{bmatrix}\overline{Y}_{\mathrm{w}}(\mathbf{x})\\ \overline{u}_{\mathrm{w}}(\mathbf{x})\\ \overline{v}_{\mathrm{w}}(\mathbf{x})\end{bmatrix} = \begin{bmatrix}\log_2 Y_{\mathrm{w}}(\mathbf{x})\\ 0.114\frac{1}{Y_{\mathrm{w}}(\mathbf{x})}B_{\mathrm{w}}(\mathbf{x})\\ 0.299\frac{1}{Y_{\mathrm{w}}(\mathbf{x})}R_{\mathrm{w}}(\mathbf{x})\end{bmatrix} \qquad Y_{\mathrm{w}}(\mathbf{x}) = \begin{bmatrix}0.299\\ 0.587\\ 0.114\end{bmatrix}^{\top}\cdot\begin{bmatrix}R_{\mathrm{w}}(\mathbf{x})\\ G_{\mathrm{w}}(\mathbf{x})\\ B_{\mathrm{w}}(\mathbf{x})\end{bmatrix} \tag{2.18}$$

where $\overline{u}$ and $\overline{v}$ are in $[0,1]$, with $\overline{u}+\overline{v}\leq 1$. Then, the image is divided into 4×4 pixels blocks. For each block, the maximum, $\overline{Y}_{\mathrm{max}}$, and the minimum, $\overline{Y}_{\mathrm{min}}$, luminance values are calculated. These values are quantised at 8-bit and stored to be used as base luminance values for the interpolation in a similar way to S3TC [87], see Appendix B for more details. Moreover, the



(a) (b)

Figure 2.8: *The encoding of chrominance in Munkberg et al. [139]: a) 2D Shapes used in the encoder. While black circles are for start and end, white circles are for interpolated values. b) An example of 2d shape fitting for a chrominance block.*

other luminance values are encoded with 2 bits, which minimise the value of the interpolation between $\overline{Y}_{\text{min}}$ and $\overline{Y}_{\text{max}}$.

At this point chrominance values are compressed, the first step is to halve the resolution of the chrominance channel. For each block, a 2D shape is chosen as the one that fits chrominance values in the $(\overline{u},\overline{v})$ plane minimising the error, see Figure 2.8. Finally, a 2 bits index is stored for each pixel which points to a sample along the fitted 2D shape.

byte	7	6	5	4	3	2	1	0
0	$\overline{Y}_{\text{max}}$							
1	$\overline{Y}_{\text{min}}$							
2	$\overline{Y}_0$		$\overline{Y}_1$		...			
	...							
10	$type_{\text{shape}}$			$\overline{u}_{\text{start}}$				
11	$\overline{u}_{\text{start}}$			$\overline{v}_{\text{start}}$				
12	$\overline{v}_{\text{start}}$	$\overline{u}_{\text{end}}$						
13	$\overline{u}_{\text{end}}$	$\overline{v}_{\text{end}}$						
14	ind_0		ind_1		ind_2		ind_3	
15	ind_4		ind_5		ind_6		ind_7	

Table 2.2: *The table shows bits allocation for a* 4×4 *block in Munkberg et al.'s method [139].*

In the decoding scheme, luminance is firstly decompressed interpolating values for each pixel in the block as:

$$\overline{Y}_{\text{w}}(\mathbf{y}) = \frac{1}{3}\left(\overline{Y}_{k(\mathbf{y})}\overline{Y}_{\text{min}} + (1-\overline{Y}_{k(\mathbf{y})})\overline{Y}_{\text{max}}\right) \tag{2.19}$$

where $\overline{Y}_{k(\mathbf{y})}$ is the $\overline{Y}$ that corresponds to a pixel at location $\mathbf{y}$. Then, the chrominance is decoded as:

$$\begin{bmatrix}\overline{u}_{\mathrm{w}}(\mathbf{y})\\ \overline{v}_{\mathrm{w}}(\mathbf{y})\end{bmatrix} = \alpha(ind_{k(\mathbf{y})})\begin{bmatrix}\overline{u}_{\mathrm{start}}-\overline{u}_{\mathrm{end}}\\ \overline{v}_{\mathrm{start}}-\overline{v}_{\mathrm{end}}\end{bmatrix} + \beta(ind_{k(\mathbf{y})})\begin{bmatrix}\overline{v}_{\mathrm{end}}-\overline{v}_{\mathrm{start}}\\ \overline{u}_{\mathrm{start}}-\overline{u}_{\mathrm{end}}\end{bmatrix} + \begin{bmatrix}\overline{u}_{\mathrm{start}}\\ \overline{v}_{\mathrm{start}}\end{bmatrix} \tag{2.20}$$

where α and β are parameters specific for each 2D shape. Subsequently, chrominance is up-sampled to the original size. Finally, the inverse $\overline{Y}\overline{uv}$ colour space transform is applied, obtaining the reconstructed pixel:

$$\begin{bmatrix}R_{\mathrm{w}}(\mathbf{x})\\ G_{\mathrm{w}}(\mathbf{x})\\ B_{\mathrm{w}}(\mathbf{x})\end{bmatrix} = 2^{\overline{Y}_{\mathrm{w}}(\mathbf{x})}\begin{bmatrix}0.229^{-1}\\ 0.587^{-1}\\ 0.114^{-1}\end{bmatrix}^{\top}\begin{bmatrix}\overline{v}_{\mathrm{w}}(\mathbf{x})\\ 1-\overline{v}_{\mathrm{w}}(\mathbf{x})-\overline{u}_{\mathrm{w}}(\mathbf{x})\\ \overline{u}_{\mathrm{w}}(\mathbf{x})\end{bmatrix} \tag{2.21}$$

The compression scheme was compared against two HDR S3TC variants using mPSNR [139], $\log_2$[RGB] RMSE [223], and HDR-VDP [127, 124]. A data set of 16 HDR textures was tested. The results showed that the method presents higher quality than S3TC variants, especially perceptually.

In conclusion the method proposed by Munkberg et al. [139] is a compression scheme that can achieve 8 bpp HDR texture compression at high quality. However, the decompression method needs special hardware, so it cannot be implemented in current graphics hardware. Furthermore, the shape fitting can take up to an hour for an one Megapixel image, which limits the scheme for fixed content.

HDR Texture Compression using Bit and Integer Operations

Co-synchronously to Munkberg et al. [139], Roimela et al. [174] presented a 8bpp BTC scheme for compressing 48 bpp HDR textures, which was later improved in Roimela et al. [175].

The first step of the coding scheme is to convert data of the texture in a colour space suitable for compression purposes. A computationally efficient colour space is defined, which splits RGB colours into luminance and chromaticity. The luminance $I_{\mathrm{w}}(\mathbf{x})$ is defined as:

$$I_{\mathrm{w}}(\mathbf{x}) = \frac{1}{4}R_{\mathrm{w}}(\mathbf{x}) + \frac{1}{2}G_{\mathrm{w}}(\mathbf{x}) + \frac{1}{4}B_{\mathrm{w}}(\mathbf{x}) \tag{2.22}$$

and chromaticity $[r_Q, b_Q]^\top$ as:

$$\begin{bmatrix} r_Q \\ b_Q \end{bmatrix} = \frac{1}{4I_\mathrm{w}(\mathbf{x})} \begin{bmatrix} R_\mathrm{w}(\mathbf{x}) \\ B_\mathrm{w}(\mathbf{x}) \end{bmatrix} \tag{2.23}$$

Then, the image is divided into 4 × 4 pixels blocks. For each block, the luminance value with the smallest bit pattern is calculated, I_Min, and its ten least significant bits are zeroed giving I_bias (only 6 bits are stored). Subsequently, I_bias is subtracted bit by bit from all luminance values in the block:

$$bit(I'_\mathrm{w}(\mathbf{y})) = bit(I_\mathrm{w}(\mathbf{y})) - bit(I_\mathrm{bias}) \tag{2.24}$$

<table>
<tr><td>byte</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>0</td><td colspan="6">I_bias</td><td colspan="2">n_zero</td></tr>
<tr><td>1</td><td>n_zero</td><td colspan="5">lum_0</td><td colspan="2">lum_1</td></tr>
<tr><td>2</td><td colspan="3">lum_1</td><td colspan="5">lum_2</td></tr>
<tr><td></td><td colspan="8">...</td></tr>
<tr><td>10</td><td colspan="4">...</td><td colspan="4">lum_{15}</td></tr>
<tr><td>11</td><td>lum_{15}</td><td colspan="6">r_bias</td><td>b_bias</td></tr>
<tr><td>12</td><td colspan="5">b_bias</td><td colspan="3">c_zero</td></tr>
<tr><td>13</td><td colspan="3">r_0</td><td colspan="3">b_0</td><td colspan="2">r_1</td></tr>
<tr><td>14</td><td>r_1</td><td colspan="3">b_1</td><td colspan="3">r_2</td><td>b_2</td></tr>
<tr><td>15</td><td colspan="2">b_2</td><td colspan="3">r_3</td><td colspan="3">b_3</td></tr>
</table>

Table 2.3: *The table shows bits allocation for a* 4 × 4 *block in Roimela et al.'s method [174].*

where *bit* operator denotes the integer bit representation of a floating point number, $\mathbf{y}$ is a pixel in the block under processing. $I'_\mathrm{w}(\mathbf{y})$ values share a number of leading zero bits that are not needed to be stored. Therefore, they are counted in the largest $I'_\mathrm{w}(\mathbf{y})$. The counter, n_zero, is clamped to seven and stored in 3 bits. At this point, the $n_\mathrm{zero} + 1$ least important bits are removed from each $I'_\mathrm{w}(\mathbf{y})$ in the block obtaining $lum_\mathrm{w}(\mathbf{y})$, which is rounded and stored at 5 bits. At this point chromaticity is compressed. Firstly, the resolution of chromaticity channels is halved. Secondly, the same compression scheme for luminance is applied to chromaticity, having two bias values at 6 bits, one for r_Q, $r_{Q,\mathrm{bias}}$, and the other for b_Q, $b_{Q,\mathrm{bias}}$. Furthermore, there is a common zero counter c_zero, and final values are rounded at 4 bits. The number of bits for luminance and chromaticity channels are respectively 88 bits and 40 bits, for a total of 128 bits or 8 bpp, see Table 2.3 for the complete allocation of bits.

The decoding scheme is quite straightforward. Firstly, luminance is decoded by bit shifting to the left each $lum_{\mathrm{y}}(\mathbf{w})$ value $n_{\mathrm{zero}}+1$ times, and adding I_{bias}. Secondly, this operation is repeated for the chromaticity channel which is subsequently up-sampled to the original size. Finally, the image is converted from Ir_Qb_Q colour space to RGB applying the inverse transform:

$$\begin{bmatrix} R_{\mathrm{w}}(\mathbf{x}) \\ G_{\mathrm{w}}(\mathbf{x}) \\ B_{\mathrm{w}}(\mathbf{x}) \end{bmatrix} = I_{\mathrm{w}}(\mathbf{x}) \begin{bmatrix} 4 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 4 \end{bmatrix} \begin{bmatrix} r_{\mathrm{Q,w}}(\mathbf{x}) \\ 1 - r_{\mathrm{Q,w}}(\mathbf{x}) - b_{\mathrm{Q,w}}(\mathbf{x}) \\ b_{\mathrm{Q,w}}(\mathbf{x}) \end{bmatrix} \tag{2.25}$$

This scheme was compared against Munkberg et al.'s scheme [139], HDR-JPEG 2000 [223], and a HDR S3TC variant using different metrics such as PSNR, mPSNR [139], HDR-VDP [127, 124], and RMSE. A data set of 18 HDR textures was tested. The results showed that the encoding method has quality similar to RGBE. Moreover, it is similar to Munkberg et al.'s scheme [139], but the chromaticity quality is lower than in that method.

In conclusion, this compression scheme presents a computationally efficient encoding/decoding scheme for 48 bpp HDR textures. Only integer and bit operations are needed. Furthermore, it achieves high quality images at only 8 bpp. However, the main drawback of the scheme is that it cannot be implemented on current hardware.

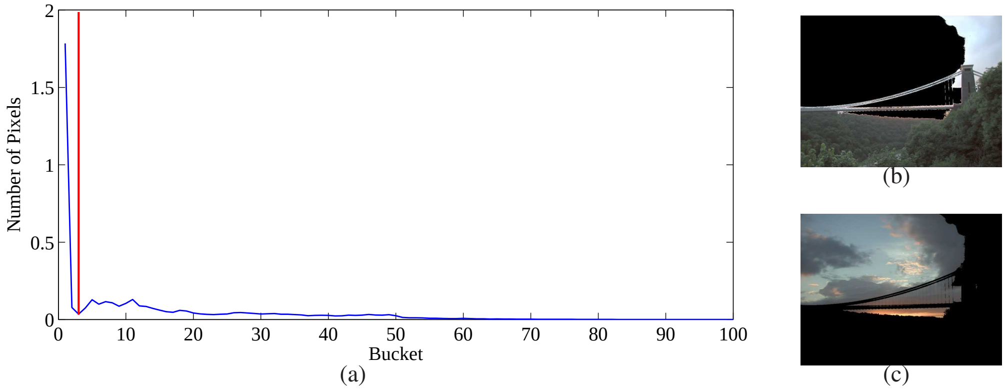
HDR Texture Compression Encoding LDR and HDR Parts

Methods such as the one of Munkberg et al. [139] and Roimela et al. [174, 175] have the main problem that they cannot be implemented on current graphics hardware. To solve this problem, Wang et al. [206] proposed a compression method based on S3TC [87]. The main idea is to split the HDR image in two parts: one with LDR values and the second part with HDR values. The two parts are stored on two S3TC textures, for a total of 16 bpp.

The encoding starts by splitting luminance and chrominance, using the $LUVW$ colour space which is defined as:

$$L_{\mathrm{w}}(\mathbf{x}) = \sqrt{R_{\mathrm{w}}(\mathbf{x})^2 + G_{\mathrm{w}}(\mathbf{x})^2 + B_{\mathrm{w}}(\mathbf{x})^2} \qquad \begin{bmatrix} U_{\mathrm{w}}(\mathbf{x}) \\ V_{\mathrm{w}}(\mathbf{x}) \\ W_{\mathrm{w}}(\mathbf{x}) \end{bmatrix} = \frac{1}{L_{\mathrm{w}}(\mathbf{x})} \begin{bmatrix} R_{\mathrm{w}}(\mathbf{x}) \\ G_{\mathrm{w}}(\mathbf{x}) \\ B_{\mathrm{w}}(\mathbf{x}) \end{bmatrix} \tag{2.26}$$

After the colour conversion, the luminance channel is split in a HDR and LDR part. This is achieved by finding the threshold, $L_{\mathrm{w,s}}$, that minimises quantisation error of encoding uniformly



(a) (b) (c)

Figure 2.9: *An example of the separation process of the LDR and HDR part in Wang et al. [206] applied to the Bristol Bridge HDR image: a) The histogram of the image, in red the axis that divides the image in LDR and HDR. b) The LDR part of the image, uniformly quantised. c) The HDR part of the image, uniformly quantised. The original HDR image is courtesy of Greg Ward [212].*

the LDR and HDR part separately, which is defined as:

$$E(L_{\mathrm{w,s}}) = n_{\mathrm{LDR}}\frac{(L_{\mathrm{w,s}})(L_{\mathrm{w,s}} - L_{\mathrm{w,\,Min}})}{2^{b_{\mathrm{LDR}}}} + n_{\mathrm{HDR}}\frac{(L_{\mathrm{w,s}})(L_{\mathrm{w,\,Max}} - L_{\mathrm{w,s}})}{2^{b_{\mathrm{HDR}}}} \tag{2.27}$$

where n_{LDR} and n_{HDR} are respectively the number of pixel in the LDR and HDR part. b_{LDR} and b_{HDR} are respectively the number of bits for quantising the HDR and LDR part. The HDR texture is stored in two S3TC textures,Tex_0 and Tex_1, as:

$$\begin{gathered}
Tex0_{\mathrm{R}}(\mathbf{x}) = U_{\mathrm{w}}(\mathbf{x}) \qquad Tex0_{\mathrm{G}}(\mathbf{x}) = V_{\mathrm{w}}(\mathbf{x}) \qquad Tex0_{\mathrm{B}}(\mathbf{x}) = W_{\mathrm{w}}(\mathbf{x}) \\
Tex0_{\mathrm{A}}(\mathbf{x}) = \begin{cases} \frac{1}{L_{\mathrm{w,s}} - L_{\mathrm{w,\,Min}}} L_{\mathrm{w}}(\mathbf{x}) & \text{if } L_{\mathrm{w,\,Min}} \le L_{\mathrm{w}}(\mathbf{x}) \le L_{\mathrm{w,s}} \\ 0 & \text{otherwise} \end{cases} \\
Tex1_{\mathrm{A}}(\mathbf{x}) = \begin{cases} \frac{1}{L_{\mathrm{w,\,Max}} - L_{\mathrm{w,s}}} L_{\mathrm{w}}(\mathbf{x}) & \text{if } L_{\mathrm{w,s}} < L_{\mathrm{w}}(\mathbf{x}) \le L_{\mathrm{w,\,Max}} \\ 1 & \text{otherwise} \end{cases}
\end{gathered} \tag{2.28}$$

where subscript $_R$, $_G$, $_B$,and $_A$ are indicating respectively the red, green, blue and alpha channel of a texture. Furthermore, additive residuals are included to improve quality. These are simply calculated as:

$$res(\mathbf{x}) = L_{\mathrm{w}}(\mathbf{x}) - L_{\mathrm{w}}(\mathbf{x})' \tag{2.29}$$

where $L_{\mathrm{w}}(\mathbf{x})'$ is the reconstructed luminance, which is defined as:

$$L_{\mathrm{w}}(\mathbf{x})' = Tex0_{\mathrm{A}}(\mathbf{x})(L_{\mathrm{w,\,Max}} - L_{\mathrm{w,s}}(\mathbf{x})) + Tex1_{\mathrm{A}}(\mathbf{x})(L_{\mathrm{w,s}}(\mathbf{x}) - L_{\mathrm{w,\,Min}}) + L_{\mathrm{w,\,Min}} \tag{2.30}$$

Then, *res* is split into three parts in a similar way as luminance and stored in the red, green and blue channel of Tex_1.

The decoding is quite straightforward. Firstly, S3TC textures are decoded, then the luminance channel is reconstructed as:

$$\begin{aligned} L_{\mathrm{w}} = L_{\mathrm{w}}(\mathbf{x})' + Tex1_{\mathrm{R}}(\mathbf{x})(res_{\mathrm{s1}} - res_{\mathrm{Min}}) + Tex1_{\mathrm{G}}(\mathbf{x})(res_{\mathrm{s2}} - res_{\mathrm{s1}}) + \\ Tex1_{\mathrm{B}}(\mathbf{x})(res_{\mathrm{Max}} - res_{\mathrm{s2}}) + res_{\mathrm{Min}} \end{aligned} \tag{2.31}$$


(a) (b) (c)

Figure 2.10: *An example of failure of the compression method of Wang et al. [206] applied to Saint Peter's Basilica HDR image: a) The image at exposure 0. b) A zoom of the red square in a) from the original image. c) A zoom of the red square in a) from the compressed image, note that quantisation artifacts are visible in the form of contouring. The original HDR image is courtesy of Paul Debevec [44].*

Finally, the image is converted from *LUVW* colour space to *RGB* colour space:

$$\begin{bmatrix} R_{\mathrm{w}}(\mathbf{x}) \\ G_{\mathrm{w}}(\mathbf{x}) \\ B_{\mathrm{w}}(\mathbf{x}) \end{bmatrix} = L_{\mathrm{w}}(\mathbf{x}) \begin{bmatrix} U_{\mathrm{w}}(\mathbf{x}) \\ V_{\mathrm{w}}(\mathbf{x}) \\ W_{\mathrm{w}}(\mathbf{x}) \end{bmatrix} \tag{2.32}$$

The compression scheme was compared against RGBE and OpenEXR formats using classic PSNR, and the quality was worse than RGBE (5-7dB less) and OpenEXR (40-60 dB less). Nevertheless, the method needs only 16 bpp compared against the 32 bpp of RGBE and 48 of OpenEXR.

In conclusion, the compression scheme presents an acceptable quality and it can be mapped on current hardware at high frame rates (470-480 fps with a 512^2 viewport and texturing for each pixel), because only simple operations are used in Equation 2.31. The main drawback of the method is that an optimal $L_{\mathrm{w,s}}$ for an image can generate quantisation artifacts that are noticeable, see Figure 2.10.

2.4 Tone Mapping

Most of the display devices available nowadays are not able to display HDR content, but only LDR content. This is due to two reasons. The first, entry level monitors/displays have a low contrast ratio around $200:1$. The second, high-end LCD televisions with a high contrast ratio, on average around $10,000:1$, cannot interpolate pixel values. This means there is still on 8-bit discretisation for each colour channel, and more information is not encoded and thus cannot be displayed. In the last two decades researchers spent significant time and effort in order to compress the range of HDR images and videos in order to be visualised on LDR displays.

Tone mapping is the operation that reduces the dynamic range of the input content to fit the dynamic range of the displaying technology. This reduction of the range has to keep some properties of the original content such as local and global contrast, details, etc. Furthermore, the perception of the tone mapped image should match the real world perception, see Figure 2.11. Tone mapping is performed using an operator f or Tone Mapping Operator (TMO), which is defined in general as:

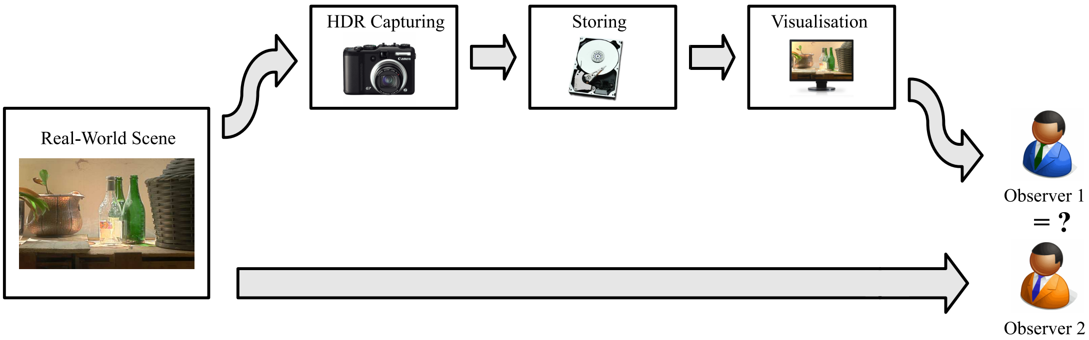


Figure 2.11: *Tone Mapping and real-world relationship. Observer 1 and Observer 2 are looking at the same scene, but in two different ways. While Observer 1 is watching it from a monitor after capturing, storing, and tone mapping, Observer 2 is watching it live. A TMO should match the perception of the real world scene.*

$$f(I): \mathbb{R}_i^{w\times h\times c} \rightarrow \mathbb{D}_o^{w\times h\times c} \tag{2.33}$$

where I is the image, w and h are respectively the width and height of I, c is the number of colour bands of I (usually $c = 3$ in RGB colour space), $\mathbb{R}_i \subseteq \mathbb{R}$, $\mathbb{D}_o \subset \mathbb{R}_i$. $\mathbb{D}_o = [0, 255]$ for normal LDR monitors. Furthermore, only luminance is usually tone mapped by a TMO while colours are unprocessed. This simplifies Equation 2.33 to:

$$f(I) = \begin{cases} L_{\text{d}} = f_{\text{L}}(L_{\text{w}}) : \mathbb{R}_i^{w \times h} \rightarrow [0, 255] \\ \\ \begin{bmatrix} R_{\text{d}} \\ G_{\text{d}} \\ B_{\text{d}} \end{bmatrix} = L_{\text{d}} \left(\frac{1}{L_{\text{w}}} \begin{bmatrix} R_{\text{w}} \\ G_{\text{w}} \\ B_{\text{w}} \end{bmatrix} \right)^s \end{cases} \tag{2.34}$$

where $s \in (0, 1]$ is a saturation factor that decreases saturation, which is usually increased during tone mapping. After the application of f, gamma correction is usually applied and each colour channel is clamped in the range $[0, 255]$.

Note that the original gamut is greatly modified in this process, and tone mapped colour appearance can result in differences from the one in the original image. Furthermore, the shape of the range compression and how details are preserved can produce distortions in global and local contrast. For instance, it can happen that details are exaggerated, introducing more local contrast than in the original image. To solve this problem, a few techniques have been recently proposed which are divided in two groups. The first one is the pre-processing techniques that modifies the original HDR image before tone mapping to avoid distortions such as Akyüz and Reinhard [8] and Yoshida et al. [227] methods. The second group is the post-processing techniques that analyse the image, determine the amount of the distortion and try to restore the tone mapped image using the HDR image and properties of the HVS. Methods such as the chroma enhancement treatment by Smith et al. [181] and the brightness enhancement by Krawczyk et al. [100] belong to this group.

TMOs can be classified in different groups based on f or the image processing techniques they use, see Table 2.4. The main groups of the taxonomy are:

- **Global Operators**: the mapping is applied to all pixels with the same operator, f.
- **Local Operators**: the mapping of a pixel depends on its neighbours, which are input for f.

	Empirical	Perceptual
Global	Linear Mapping [178] Exponential and Logarithmic Mapping [178] Quantisation Techniques [178]	Perceptual Brightness Reproduction^{T} [193] Contrast Based Scale Factor [210] A model of Visual Adaptation [63] Histogram Adjustment [107] Time Dependent Visual Adaptation^{T} [156] Adaptive Logarithm [52]
Local	A Spatially Variant Tone Reproduction[31] Photographic Tone Reproduction [169]	Multi-scale observer model [155] Multi-scale TVI [10] A local Model of Eye Adaptation^{T} [109] Retinex Methods [165, 134]
Frequency	LICS [195] Gradient Domain Compression [61]	Fast Bilateral Filtering [55] Trilateral Filtering [32] iCAM 2006 [101]
Segmentation	Interactive Manipulation [118] Exposure Fusion [131]	Segmentation [224] Lightness Perception [99]

Table 2.4: *The taxonomy of TMOs, which are divided based on their image processing techniques and their f. Upperscript ^{T} means that the operator is temporal and suitable for HDR videos.*

- **Segmentation Operators**: the image is segmented in broad regions, and a different mapping is applied to each region.
- **Frequency/Gradient Operators**: low and high frequencies of the images are separated, while an operator is applied to the low frequencies, high frequencies are usually kept as they are to preserve fine details.
- **Perceptual Operators**: these operators can be Global or Local or Segmentation or Frequency/Gradient. The main focus is the function f, that models some aspects of the HVS.
- **Empirical Operators**: these operators can be Global or Local or Segmentation or Frequency/Gradient. In this case, f does not try to mimic the HVS, but it tries to create pleasant images inspired by other fields such as photography.
- **Temporal Operators**: these operators are designed to be suitable for tone mapping HDR videos.

In the next subsections, the main TMOs are reviewed. This review is organised by image processing techniques which are: global operators in Section 2.4.1, local operators in Section 2.4.2, frequency operators in Section 2.4.3, and segmentation operators in Section 2.4.4.

2.4.1 Global Operators

In global operators, the same operator f is applied to all pixels of the image, with the goal to preserve global contrast. The operator can have a first pass in which some statistics of the image are calculated to optimise the dynamic range reduction. More common statistics for tone mapping are: maximum luminance, minimum luminance, and geometric or arithmetic average values. To increase robustness and to avoid outliers these statistics are calculated using percentiles, especially for minimum and maximum values, because they can be affected by noise during image capture. These operators are straightforward to extended in the temporal domain, indeed statistics are needed only to be temporally filtered to take care of flickering artifacts. The main drawback of global operators is that they usually cannot keep local contrast and fine details of the original HDR image.

Simple Mapping Methods

Simple operators are based on basic functions such as linear scaling, logarithm and exponential. While they are usually fast and simple to implement, they cannot compress the dynamic range much. The main simple operators are: single exposure or linear mapping, logarithmic mapping, and exponential mapping.

Single exposure is a straightforward way to visualise HDR images. The starting image is multiplied by e:

$$L_{\mathrm{d}}(\mathbf{x}) = eL_{\mathrm{w}}(\mathbf{x}) \tag{2.35}$$

e is chosen by the user based on information that is interesting to visualise. When $e = L_{\mathrm{w,\,Max}}$ this scaling is called normalisation and it can cause a very dark appearance, see Figure 2.12.a for an example. If e is calculated by maximising the number of well-exposed pixel as in [142], the scaling is called automatic exposure, for an example see Figure 2.12.b. Ward [210]

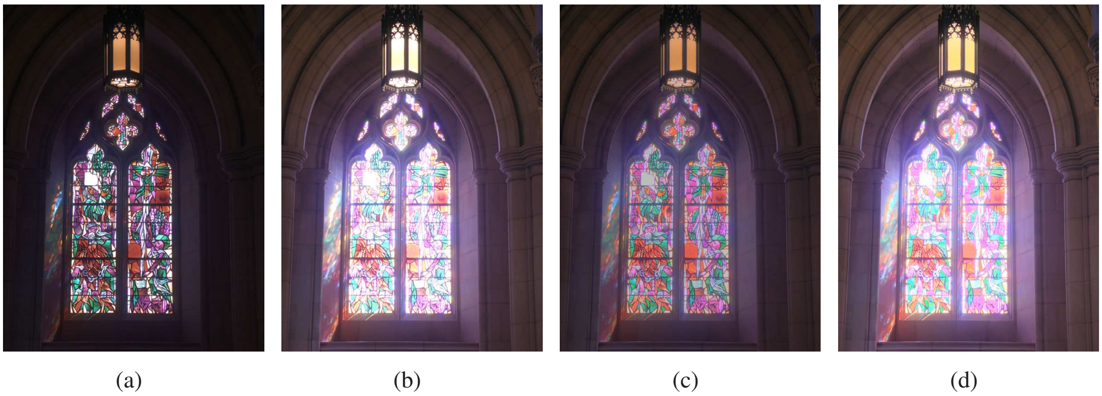


Figure 2.12: *An example of the applications of simple operators to Cathedral HDR image: a) Normalisation. b) Automatic exposure. c) Logarithmic mapping $q = 0.01$ and $k = 1$. d) Exponential Mapping $q = 0.1$ and $k = 1$. The original image is courtesy of Max Lyons [121].*

proposed to calculate e by matching the threshold visibility in an image and a display using threshold-versus-intensity (TVI) functions. However, a simple linear scale cannot compress the dynamic range of the scene, hence it shows only a slice of information.

Logarithm mapping applies a logarithm function to HDR values. The base is the maximum value of the HDR image to map non-linear values in the range $[0,1]$. The operator is defined as:

$$L_{\mathrm{d}}(\mathbf{x}) = \frac{\log_{10}\left(1 + qL_{\mathrm{w}}(\mathbf{x})\right)}{\log_{10}\left(1 + kL_{\mathrm{w,max}}\right)} \tag{2.36}$$

where $q \in [1, \infty)$ and $k \in [1, \infty)$ are constants selected by the user for determining the desired appearance of the image.

Exponential mapping applies an exponential function to HDR values. It re-maps values in the interval $[1/e, 1]$, where each value is divided by the arithmetic average. The operator is defined as:

$$L_{\mathrm{d}}(\mathbf{x}) = 1 - \exp\left(\frac{qL_{\mathrm{w}}(\mathbf{x})}{kL_{\mathrm{w,\,Avg}}}\right) \tag{2.37}$$

where $q \in [1, \infty)$ and $k \in [1, \infty)$ are constants selected by the user for determining the desired appearance of the image.

Both exponential and logarithmic mapping can deal with medium dynamic range content, but their functions cannot compress HDR content well, causing effects in images/videos such as very dark/bright appearance, low preservation of global contrast, an unnatural look, etc.

Brightness Reproduction

One of the first TMOs in the field of computer graphics was proposed by Tumblin and Rushmeier [194]. This was revised by Tumblin et al. [193]. Their operator is inspired by the HVS and more in particular by Stevens and Stevens's work on brightness [184, 183]. The TMO is defined as:

$$L_{\mathrm{d}}(\mathbf{x}) = mL_{\mathrm{da}}\left(\frac{L_{\mathrm{w}}(\mathbf{x})}{L_{\mathrm{w,\,H}}}\right)^{\alpha} \qquad \alpha = \frac{\gamma(L_{\mathrm{w,\,H}})}{\gamma(L_{\mathrm{da}})} \tag{2.38}$$

where L_{da} is the adaptation luminance (30-100 cd/m^2 for LDR displays). $\gamma(x)$ is the Stevens and Stevens' contrast sensitivity function for a human adapted to a luminance value x, which is defined as:

$$\gamma(x) = \begin{cases} 1.855 + 0.4\log_{10}\left(x + 2.3\cdot 10^{-5}\right) & \text{for } x \leq 100 \text{ cd/m}^2 \\ 2.655 & \text{otherwise} \end{cases} \tag{2.39}$$

Finally, m is the adaptation-dependent scaling term which prevents anomalous grey night images, and it is defined as:

$$m = C_{\max}^{\frac{\gamma_{\mathrm{wd}}-1}{2}} \qquad \gamma_{\mathrm{wd}} = \frac{\gamma(L_{\mathrm{w,\,H}})}{1.855 + 0.4\log_{10}\left(L_{\mathrm{da}}\right)} \tag{2.40}$$

where $C_{\max}$ is the maximum contrast that the display can visualise.

This operator compresses HDR images preserving brightness, producing plausible results when calibrated luminance values are available. Note that the TMO needs gamma correction to avoid dark images even if it includes a power function, see Figure 2.13.

Quantisation Techniques

A simple operator based on rational functions was proposed by Schlick [178], which aims to provide a simple and intuitive approach. The TMO is defined as:

(a) (b)

Figure 2.13: *An example of Tumblin and Rushmeier's operator applied to the Bottles HDR image: a) The tone mapped image without gamma correction. b) The tone mapped image with* $\gamma = 2.2$.

$$L_{\mathrm{d}}(\mathbf{x}) = \frac{pL_{\mathrm{w}}(\mathbf{x})}{(p-1)L_{\mathrm{w}}(\mathbf{x}) + L_{\mathrm{w,\ Max}}} \tag{2.41}$$

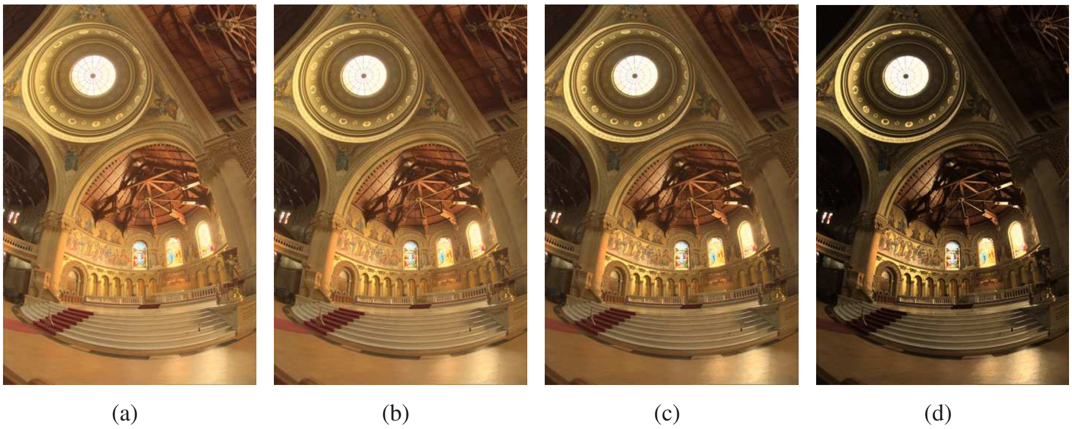
where $p \in [1, \infty)$ determines the contrast, and it can be automatically estimated as:

$$p = \frac{L_0 L_{\mathrm{w,\ Max}}}{2^N L_{\mathrm{w,\ Max}}} \tag{2.42}$$

where N is the number of bits of the output image, L_0 is the lowest luminance value that can be noticed in a monitor by an user. The use of an automatic p as in Equation 2.42 is an uniform quantisation process, because the same function is applied to all pixels. A non uniform process can be adopted using a spatially varying p for determining for each pixel of the image a local adaptation:

$$p(\mathbf{x}) = \frac{pL_{\mathrm{w}}(\mathbf{x})}{(p-1)L_{\mathrm{w}}(\mathbf{x}) + L_{\mathrm{w,\ Max}}} \left(1 - k + k\frac{L_{\mathrm{w}}(\mathbf{x})}{\sqrt{L_{\mathrm{w,max}} L_{\mathrm{w,min}}}}\right) \tag{2.43}$$

where $k \in [0, 1]$ is a weight of non-uniformity which is chosen by the user. In conclusion, the quantisation techniques provide a simple and computationally fast TMO. However, user interaction is needed to specify the appropriate k value for each image, see Figure 2.14.



(a) (b) (c) (d)

Figure 2.14: *An example of quantisation techniques applied to the Stanford Memorial Church HDR image: a) Uniform technique using automatic estimation for p, Equation 2.42. b) Non uniform technique with* $k = 0.33$*. c) Non uniform technique with* $k = 0.66$ *d) Non uniform technique with* $k = 0.99$*. The original HDR image is courtesy of Paul Debevec [47].*

A Model of Visual Adaptation

Another TMO based on psychophysical experiments was proposed by Ferwerda et al. [63] and then improved and extended by Durand and Dorsey [54]. The aim of the proposed operator is to model some aspects of the HVS such as the changes in threshold visibility, colour appearance, visual acuity, and sensitivity over time which depends on adaptation mechanisms of the HVS, see Figure 2.15 for an example. This is achieved by using TVI functions for modelling rods, m_r, (photopic vision) and cones, m_c, (scotopic vision). The operator is a simple linear scale m_c of each colour channel, added by an achromatic term scaled by m_r. While m_c and the colour channel simulate photopic conditions, m_r and the achromatic term simulate scotopic conditions. The operator is given by:

$$\begin{bmatrix} R_{\text{d}}(\mathbf{x}) \\ G_{\text{d}}(\mathbf{x}) \\ B_{\text{d}}(\mathbf{x}) \end{bmatrix} = m_c(L_{\text{da}}, L_{\text{wa}}) \begin{bmatrix} R_{\text{w}}(\mathbf{x}) \\ G_{\text{w}}(\mathbf{x}) \\ B_{\text{w}}(\mathbf{x}) \end{bmatrix} + m_r(L_{\text{da}}, L_{\text{wa}}) \begin{bmatrix} L_{\text{w}}(\mathbf{x}) \\ L_{\text{w}}(\mathbf{x}) \\ L_{\text{w}}(\mathbf{x}) \end{bmatrix} \tag{2.44}$$

where L_{wa} and L_{da} are respectively the world and display luminance adaptation. Ferwerda et al. suggested to set them equal $L_{\text{w,max}}$ for L_{wa}, and equal to half of the display maximum luminance for L_{da}.

Durand and Dorsey [54] extended the operator for working in the mesopic vision based on work of Walraven and Valeton [204]. Moreover, a time depending mechanism was introduced

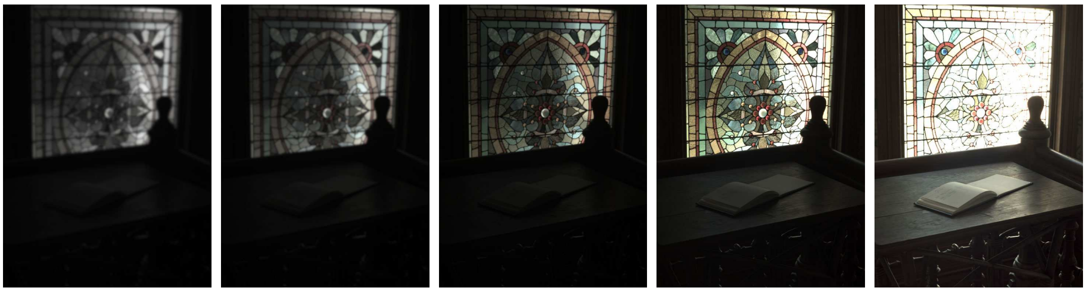

(a) (b) (c) (d) (e)

Figure 2.15: *An example of the operator proposed by Ferwerda et al. [63] of the Desk HDR image, varying mean luminance of the image: a) 0.01 cd/m^2. b) 0.1 cd/m^2. c) 1 cd/m^2. d) 10 cd/m^2. e) 100 cd/m^2. The original HDR image is courtesy of Industrial of Light and Magic from OpenEXR SDK [86].*

using data from Adelson [2] and Hayhoe [78]

The TMO proposes the simulation of many aspects of the HVS, but the reduction of the dynamic range is achieved through a simple linear scale which cannot compress the dynamic range much.

Histogram Adjustment

The classic technique of histogram equalisation [69] was modified and applied to tone mapping problem by Ward et al. [107] including aspects of HVS such as glare, loss of acuity, and colour sensitivity.

Firstly, the operator calculates the histogram of the image, f, in $\log_2$ space, using a number of bins n_{bin}. Ward et al. found out that 100 bins are adequate for accurate results. At this point, the cumulative histogram P is computed as:

$$P(x) = \sum_{i=1}^{x} \frac{f(i)}{T} \qquad T = \sum_{i=1}^{n_{\text{bin}}} f(i) \tag{2.45}$$

where x is a bin. Note that the cumulative histogram is an integration, while the histogram is its derivative with an appropriate scale:

$$\frac{\partial P(x)}{\partial x} = \frac{f(x)}{T\Delta x} \qquad \Delta x = \frac{\lceil \log(L_{\text{w, Max}}/L_{\text{w, Min}}) \rceil}{n_{\text{bin}}} \tag{2.46}$$

Then, the histogram needs to be equalised. A classic equalisation contrast such as:

$$\log(L_{\mathrm{d}}(\mathbf{x})) = \log(L_{\mathrm{d,min}}) + P(\log L_{\mathrm{w}}(\mathbf{x}))\log(L_{\mathrm{d,max}}/L_{\mathrm{d,min}}) \tag{2.47}$$

exaggerates contrast in large areas of the images, due to compression of the range in areas with few samples, and expansion in very populated ones, see Figure 2.16.a . This problem is solved by limiting the contrast enhancement. This can be expressed as:

$$\frac{\partial L_{\mathrm{d}}}{\partial L_{\mathrm{w}}} \leq \frac{L_{\mathrm{d}}}{L_{\mathrm{w}}} \tag{2.48}$$

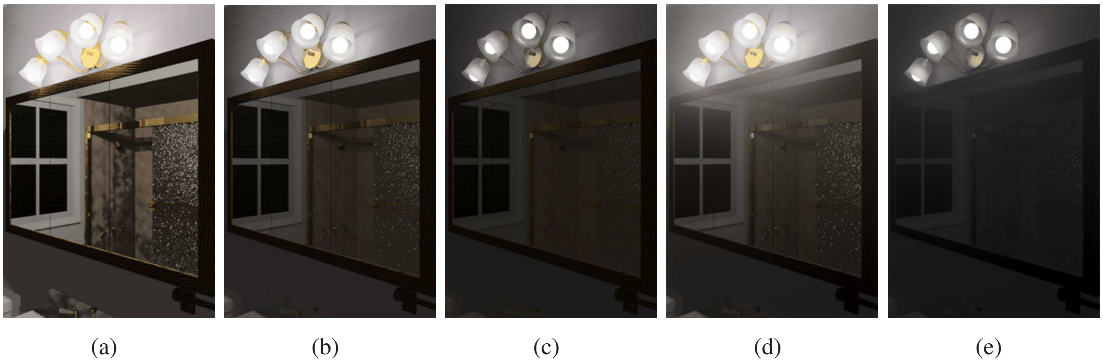

(a) (b) (c) (d) (e)

Figure 2.16: *The various stage of Histogram adjustment by Ward et al. [107] applied to the Bathroom HDR image: a) The classic histogram equalisation. b) The histogram equalisation using truncation. c) Simulation of human contrast. d) Simulation of glare. e) The final image with all effects. Images are courtesy of Greg Ward [107].*

The differentiation of Equation 2.47, using Equation 2.46, and applying Equation 2.48 leads to:

$$e^{L_{\mathrm{d}}}\frac{f(L_{\mathrm{w}})}{T\Delta x}\frac{\log(L_{\mathrm{d,max}}/L_{\mathrm{d,min}})}{L_{\mathrm{w}}} \leq \frac{L_{\mathrm{d}}}{L_{\mathrm{w}}} \tag{2.49}$$

which is reduced to a condition on $f(x)$:

$$f(x) \leq c \qquad \text{where } c = \frac{T\Delta x}{\log(L_{\mathrm{d,max}}/L_{\mathrm{d,min}})} \tag{2.50}$$

This means that exaggeration happens when Equation 2.50 is not satisfied. A solution is to truncate $f(x)$, which has to be done iteratively to avoid changes in T and subsequently changes in c. An example of this is shown in Figure 2.16.b. The operator introduces some mechanism to mimic the HVS such as limitation of the contrast, Figure 2.16.c, acuity, Figure 2.16.d, and colour sensitivity. These are in part inspired by Ferwerda et al. [63], for more details see the original paper.

In summary, the operator presents a modified histogram equalisation for HDR images which achieves good range compression and overall contrast simulating some aspects of the HVS.

Time-Dependent Visual Adaptation

The HVS adapts to large changes in luminance intensity that happens in a scene. This adaptation process is not immediate, but it takes time that is proportional to the amount of variation in the luminance levels. Pattanaik et al. [156] proposed a time dependent TMO to take into account this important effect of the HVS, which can process images and videos. Furthermore, the method is more general than Ferwerda et al.'s one [63], and in accordance with the results in psychophysics, physiology, and colour science.

The first step of the TMO is to convert the RGB input image into luminance values for rods, L_{rod}, and cones, L_{cone}, see Figure 2.17 for the complete pipeline. Then, the response RS of rods and cones is calculated following Hunt's model [84]:

$$RS_{\text{rod}}(\mathbf{x}) = B_{\text{rod}} \frac{L_{\text{rod}}(\mathbf{x})^n}{L_{\text{rod}}(\mathbf{x})^n + \sigma_{\text{rod}}^n} \qquad RS_{\text{cone}}(\mathbf{x}) = B_{\text{cone}} \frac{L_{\text{cone}}(\mathbf{x})^n}{L_{\text{cone}}(\mathbf{x})^n + \sigma_{\text{cone}}^n} \tag{2.51}$$

where $n = 0.73$ [198], σ is the half-saturation parameter. B is the bleaching parameter defined as:

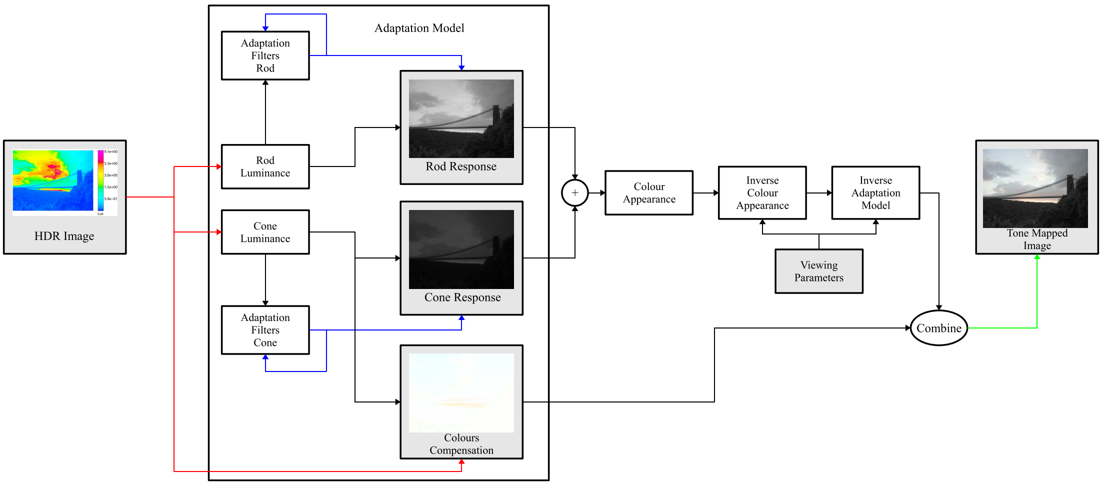


Figure 2.17: *The pipeline of the adaptation operator by Pattanaik et al. [156].*

$$B_{\text{cone}} = \frac{2\times10^6}{2\times10^6 + G_{\text{cone}}} \qquad B_{\text{rod}} = \frac{0.004}{0.004 + G_{\text{rod}}} \tag{2.52}$$

where G_{cone} and G_{rod} are parameters at adaptation time for a particular luminance value. To have a dynamic model G_{cone} and G_{rod} need to be time dependent, $G_{\text{cone}}(t)$ and $G_{\text{rod}}(t)$. Firstly, the steady state G_{cone} and G_{rod} are computed as one fifth of the reference white. Secondly, the time dependency is modelled using two exponential filters with output feedback, $1 - e^{\frac{t}{t_0}}$ where $t_{0,\text{rod}} = 150\text{ms}$ and $t_{0,\text{cone}} = 80\text{ms}$. Note that colours are simply modified to take into account range compression in Equation 2.51, as:

$$\begin{bmatrix} R'(\mathbf{x}) \\ G'(\mathbf{x}) \\ B'(\mathbf{x}) \end{bmatrix} = \left(\frac{1}{L_{\text{cone}}(\mathbf{x})} \begin{bmatrix} R_{\text{w}}(\mathbf{x}) \\ G_{\text{w}}(\mathbf{x}) \\ B_{\text{w}}(\mathbf{x}) \end{bmatrix} \right)^{S} \qquad S(\mathbf{x}) = \frac{n B_{\text{cone}} L_{\text{cone}}(\mathbf{x})^n \sigma_{\text{cone}}^n}{(L_{\text{cone}}(\mathbf{x})^n + \sigma_{\text{cone}}^n)^2} \tag{2.53}$$

After the computation of RS_{rod} and RS_{cone}, a colour appearance model (CAM) [58] is applied to $RS_{\text{lum}} = RS_{\text{rod}} + RS_{\text{cone}}$. This model simply subtracts the reference black response $\text{Ref}_{\text{black}}$ to RS_{lum}. Furthermore, the reference black is calculated as $\frac{1}{32}$ the intensity of reference white, $\text{Ref}_{\text{white}} = 5(A_{\text{rod}} + A_{\text{cone}})$.

The final step is to inverse the appearance model and the adaptation model to match the viewing condition in an ordinary office lighting using an LDR display.

Figure 2.18: *A comparison between the method original method by Pattanaik et al. [156] in the top row, and Irawan et al. [88] in the bottom row, using the Tunnel sequence. Images are courtesy of Sumanta Pattanaik [156] and Jim Ferwerda [88].*

The operator is a simple global time dependent and fully automatic TMO which lacks some secondary effects of the HVS (glare, acuity, etc.), for an example see Figure 2.18. This model was extended by Ledda et al. [109], proposing a local model and fully automatic of visual adaptation for static images and videos. Additionally, they presented a psychophysical validation study of the TMO using a HDR Monitor [179]. Their results showed a strong correlation between tone mapped images displayed on a LDR monitor and linear HDR images showed on the HDR one. A further extension of Pattanaik et al. [156] operator was proposed by Irawan et al. [88] who combined the method with the histogram adjustment operator by Ward et al. [107], see Figure 2.18. Their work simulates visibility in time-varying, high dynamic range scenes for observers with low vision. This is achieved using a temporally coherent histogram adjustment method combined with an adaptive TVI function based on the measurements of Naka and Rushton [140].

Adaptive Logarithmic Mapping

A global mapping based on the logarithm was proposed by Drago et al. [52]. The novelty compared to the range reduction using a simple logarithm is that the capacity to adapt the logarithm base for each pixel of the image is inspired by the HVS. The brightness, which is the response of the HVS to a luminance stimuli, presents a non-linear response. This can be approximated using a logarithm, the Weber-Fechner Law, given by:

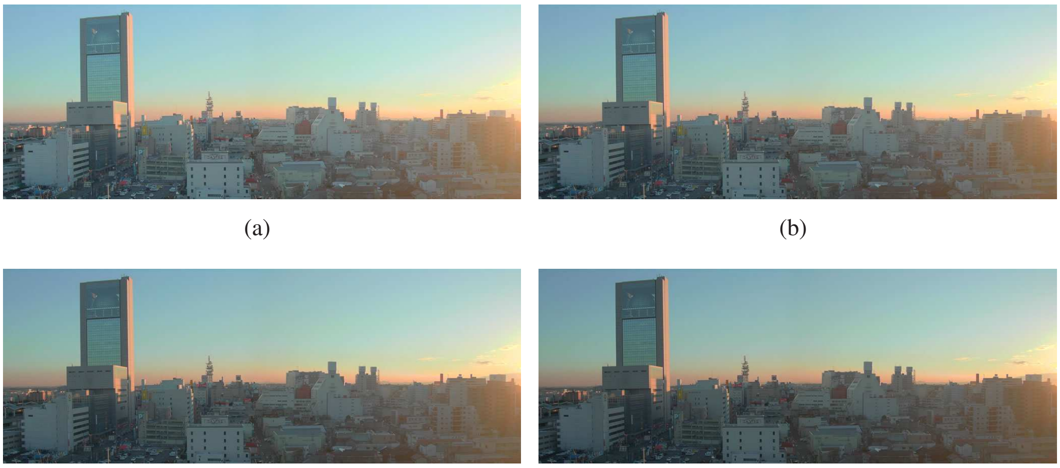
$$B = k_1 \log\left(\frac{L}{L_0}\right) \tag{2.54}$$

where L is the stimuli's luminance, L_0 the background's luminance, and k_1 is a constant.

Following the Weber-Fechner Law, the mapping is defined as:

$$L_{\mathrm{d}}(\mathbf{x}) = \frac{L_{\mathrm{d,max}}}{100\log_{10}(1+L_{\mathrm{w,max}})} \frac{\log_{10}(1+L_{\mathrm{w}}(\mathbf{x}))}{\log_{10}\left(2+8\left(\frac{L_{\mathrm{w}}(\mathbf{x})}{L_{\mathrm{w,max}}}\right)^{\alpha}\right)} \qquad \alpha = \log_{0.5} p \tag{2.55}$$

where $p \in [0,1]$ is a user parameter that controls contrast appearance suggested $p = 0.85$, and $L_{\mathrm{d,max}} = 100\mathrm{cd/m}^2$ which is a common maximum luminance value for LDR displays, see



(a) (b)

(c) (d)

Figure 2.19: *An example of the adaptive logarithmic mapping applied to the Koriyama City HDR image, varying p: a)* $p = 0.65$ *b)* $p = 0.75$ *c)* $p = 0.85$ *d)* $p = 0.95$. *The original HDR image is courtesy of Karol Myszkowski.*

Figure 2.19. Note that the operator applies an adaptive logarithm based on luminance which varies from $\log_2$ for dark regions for keeping contrast and visibility, to $\log_{10}$ for bright regions to reinforce contrast compression.

The TMO is a fast mapping that allows a good global range compression. However, its global nature does not allow the preservation of fine details.

2.4.2 Local Operators

Local operators improve the quality over global operators, because they try to reproduce both local and global contrast. This is achieved by including in f, the mapping operator, intensity values from the neighbouring pixels of the one that is going to be tone mapped. However, neighbours have to be chosen carefully, otherwise halos around edges can appear. Halos are sometimes desired when attention needs to be given to a particular area, but if the phenomenon is uncontrolled it can produce unpleasant images.

Spatially Non-uniform Scaling

Chiu et al. [31] proposed the first operator with the attempt to preserve the local contrast. The TMO scales the world luminance by the average of neighbouring pixels, which is defined as:

$$L_{\mathrm{d}}(\mathbf{x}) = L_{\mathrm{w}}(\mathbf{x})s(\mathbf{x}) \tag{2.56}$$

where $s(\mathbf{x})$ is the local average of neighbour pixels, defined as:

$$s(\mathbf{x}) = \big(k\big(L_{\mathrm{w}} \otimes G_{\sigma})(\mathbf{x})\big)^{-1} \tag{2.57}$$


(a) (b) (c) (d)

Figure 2.20: *An example of the local TMO introduced by Chiu et al. [31] applied to the Stanford Memorial Church HDR image: a) The simple operator with $\sigma = 3$. While local contrast is preserved, the global contrast is completely lost resulting in a flat appearance of the tone mapped image. b) The simple operator with $\sigma = 27$. In this case both local and global contrast are kept, but halos are quite extended in the image. c) The TMO with clamping and $\sigma = 27$. Halos are reduced but not completely removed. d) The full TMO with glare simulation and $\sigma = 27$, note that glare masks halos. The original HDR image is courtesy of Paul Debevec [47].*

where G_σ is a Gaussian filter, and k is a constant which scales the final output. The main problem of the operator is that while a small σ value produces a very low contrast image, see Figure 2.20.a, a high σ value generates halos in the image, see Figure 2.20.b. Halos are caused at edges between a very bright area and a very dark one, which means that $s(\mathbf{x}) > L_{\mathrm{w}}(\mathbf{x})^{-1}$. To alleviate halos, a pixel values is clamped to $L_{\mathrm{w}}(\mathbf{x})^{-1}$ if $s(\mathbf{x}) > L_{\mathrm{w}}(\mathbf{x})^{-1}$. At this point s can still have artifacts in the form of steep gradients where $s(\mathbf{x}) = L_{\mathrm{w}}(\mathbf{x})^{-1}$. A solution is to smooth s iteratively with a 3×3 Gaussian filter, see Figure 2.20.d. Finally, the operator masks the remaining halos artifacts simulating glare which is modelled by a low-pass filter with a square root shape.

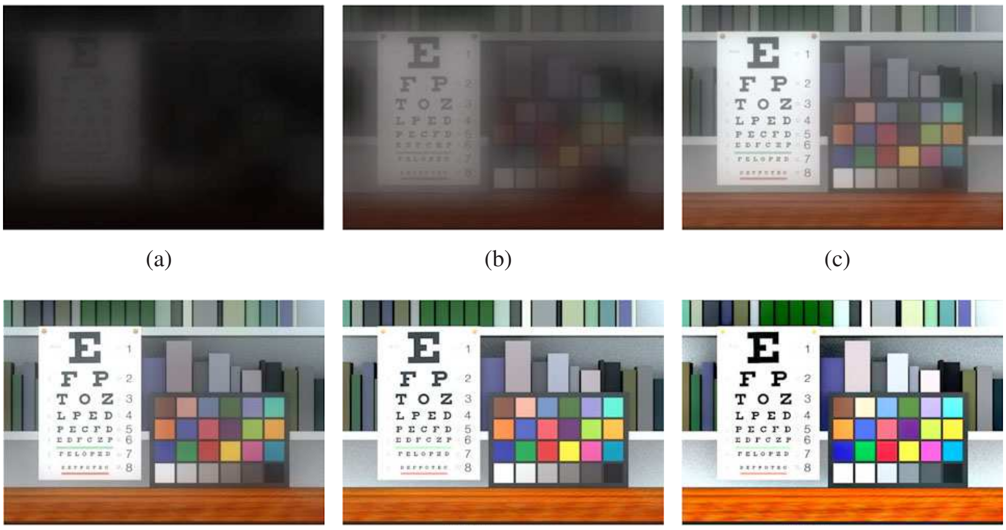
The operator presents the first local solution, however it is quite computationally expensive for alleviating halos (around 1000 iterations for the smoothing step), there are many parameters which need to be tuned, and finally halos are reduced but not completely removed by the clamping, smoothing step, and glare.

Multi-scale Observer Model

Pattanaik et al. [155] proposed the first spatially varying colour appearance model (CAM) for the reproduction of HDR images. This CAM is based on previous psychophysical works and experiments arranged in a coherent framework, see the original paper for more details. The key fact, from these previous works, is that the visual processing can be described by filtering at different scales [222].

The algorithm processes images in the Long, Middle, and Short (LMS) wavelength cones responses and rods response (R). The cones and rods responses are processed spatially. Firstly, the image is decomposed into a stack of seven Gaussian filtered images. Secondly, for each channel a stack of Difference of Gaussian (DoG) images is computed. Gain control functions for cones and rods are applied at each level which converts the contrast to adapted contrast. Then, a transducer function, T is applied to each adapted DoG to model psychophysically derived human spatial contrast sensitivity functions as described in Watson and Solomon [218]. Finally, chromatic and achromatic cones are calculated following the Hunt Model [84]. In order to map the model on a LDR display, T and G are inverted for the viewing condition of a the target display. Subsequently, they are applied respectively to the LMS components of the image, first T and then G. Finally, the image is reconstructed combining the stack of images. Note that, the range reduction is performed by function G and T.

The multi-scale observer model is a CAM model designed for dynamic range reduction. The model, due to its local nature, can simulate many aspects of the HVS such as visual acuity, change in threshold visibility, colour discrimination, and colourfulness, see Figure 2.21. However, halos can occur if the starting kernel for the decomposition is not chosen accurately. Furthermore, the model is computationally expensive due to the high number of filtering operations and intermediate states of processing that are needed. Similar algorithms to the multi-scale observer model are multi-scale methods based on Retinex theory [104], such as Raham's work [165] and extension to HDR imaging [134].



(a) (b) (c)

(d) (e) (f)

Figure 2.21: *An example of the multi-scale observer model by Pattanaik et al. [155] applied to a scene with the Macbeth chart HDR image varying the mean luminance: a) 0.1 cd/m*^{2}*. b) 1 cd/m*^{2}*. c) 10 cd/m*^{2}*. d) 100 cd/m*^{2}*. e) 1000 cd/m*^{2}*. f) 10000 cd/m*^{2}*. Images are courtesy of Sumanta Pattanaik [155].*

Photographic Tone Reproduction

A local operator based on photographic principles was presented by Reinhard et al. [169]. The idea is to simulate the burning and dodge effect that photographers have applied for more than a century, in particular the operator is inspired by the Zonal System presented by Adams [1].

The local operator is based on a global one, which is a simple sigmoid defined as:

$$L_{\mathrm{d}}(\mathbf{x}) = \frac{L_{\mathrm{m}}(\mathbf{x})}{1 + L_{\mathrm{m}}(\mathbf{x})} \tag{2.58}$$

where L_{m} is the original luminance scaled by $aL_{\mathrm{w,\,H}}^{-1}$, and a is the chosen exposure for developing the film in the photographic analogy. Note that in Equation 2.58, while high values are compressed, others are scaled linearly. However, Equation 2.58 does not allow bright areas to be burn out, as a photographer could do during the developing of a film for enhancing contrast. Therefore, Equation 2.58 can be modified as follows:

$$L_{\mathrm{d}}(\mathbf{x}) = \frac{L_{\mathrm{m}}(\mathbf{x})\left(1 + L_{\mathrm{white}}^{-2} L_{\mathrm{m}}(\mathbf{x})\right)}{1 + L_{\mathrm{m}}(\mathbf{x})} \tag{2.59}$$

where L_{white} is the smallest luminance value that is mapped to white, and is equal to $L_{\text{m, Max}}$ by default. If $L_{\text{white}} < L_{\text{m, Max}}$, values which are greater than L_{white} are clamped (burnt in the photography analogy).


(a) (b)

Figure 2.22: *An example of photographic tone reproduction operator by Reinhard et al. [169] applied to the Bottles HDR image: a) Global operator from Equation 2.58 with* $a = 0.1$. *b) Local operator from Equation 2.58 with* $a = 0.1$. *Note that edges are preserved, especially in reflections on bottles.*

A local operator can be defined for Equation 2.58 and Equation 2.59. This is achieved by finding the largest local area without sharp edges for avoiding halo artifacts. This area can be detected by comparing different size Gaussian filtered L_{m} images. If the difference is very small or tends to zero there is no edge, otherwise there is. The comparison is defined as:

$$\left| \frac{L_{\sigma}(\mathbf{x}) - L_{\sigma+1}(\mathbf{x})}{2^{\Phi} a \sigma^{-2} + L_{\sigma}(\mathbf{x})} \right| \leq \varepsilon \tag{2.60}$$

where $L_{\sigma}(\mathbf{x}) = (L_{\text{m}} \otimes G_{\sigma})(\mathbf{x})$ is a Gaussian filtered image at scale σ, ε is a small value greater than zero. Note that the difference of Gaussian filtered images is normalised to be independent by absolute values, the term $2^{\Phi} a \sigma^{-2}$ avoids singularities, and Φ is proportional to the sharpening. Once the largest σ or σ_{Max}, that satisfies Equation 2.60, is calculated for each pixel, the global operators can be modified to be local. For example Equation 2.58 is modified as:

$$L_{\text{d}}(\mathbf{x}) = \frac{L_{\text{m}}(\mathbf{x})}{1 + L_{\sigma_{\text{Max}}}(\mathbf{x})} \tag{2.61}$$

and similarly for Equation 2.59. A comparison of the two TMOs in the global and local version can be seen in Figure 2.22.


(a) (b)

Figure 2.23: *A comparison between the TMO by Reinhard et al. [169] and the one by Ashikhmin [10] applied to the Bottles HDR image: a) The local operator of Reinhard et al. [169]. b) The local operator of Ashikhmin [10]. Note that both in a) and b) details are preserved similarly, the main difference is in the global tone function.*

The photographic tone reproduction operator is a local operator that preserves edges, avoiding halo artifacts. Another advantage is that it does not need to have calibrated images as an input. Furthermore, parameters can be estimated automatically [168]. An operator which has a similar mechanism for preserving edges and avoiding halos was proposed by Ashikhmin [10]. The main difference between the two TMOs is that Ashikhmin's operator is based on the HVS for the range reduction without modelling other effects. The range reduction function, based on a TVI, takes as input a spatially varying world adaptation, $L_{\mathrm{wa}}(\mathbf{x})$ which is calculated similarly to $L_{\sigma,\max}(\mathbf{x})$ in Equation 2.61, see Figure 2.23.

2.4.3 Frequency Based Operators

Frequency based operators have the same goal as local ones, to preserve edges and local contrast, but this is achieved by a different kind of processing. These operators work in the frequency domain instead of the spatial domain. The basic observation is that edges and local contrast is preserved if and only if low frequencies are tone mapped.

Fast Bilateral Filtering Operator

The bilateral filter, see Appendix A, can separate an image into a high frequency image, detail layer, and a low frequency image with preserved edges, base layer. Durand and Dorsey

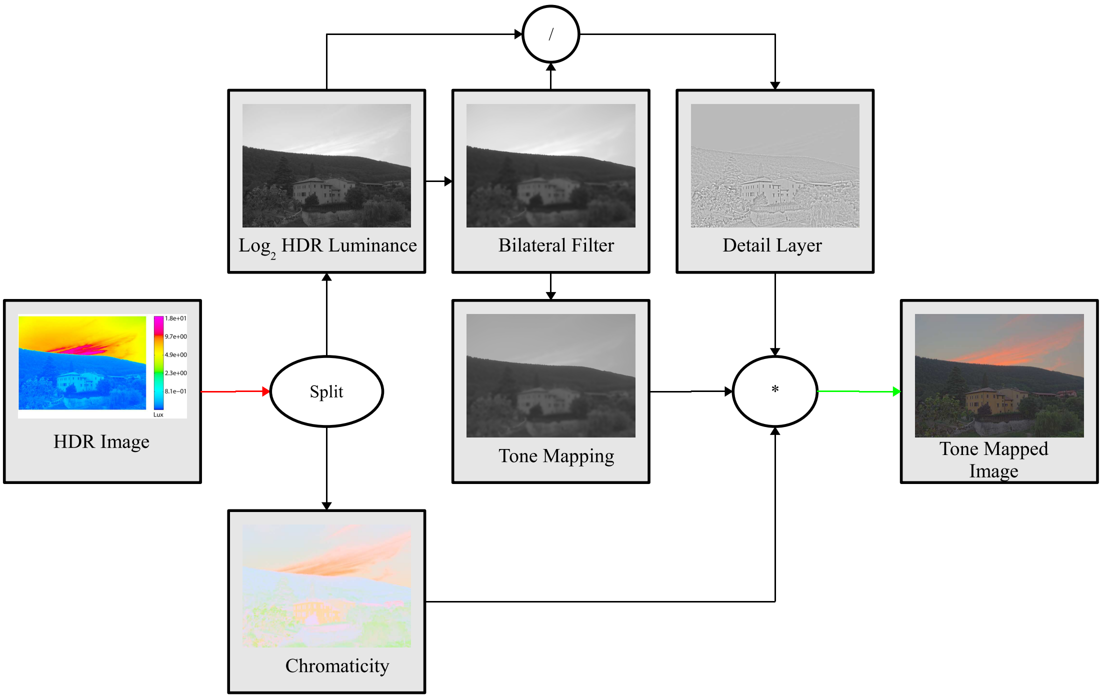


Figure 2.24: *The pipeline of the fast bilateral filtering operator.*

exploited this property for designing a general and efficient tone mapping framework which preserves local contrast and is inspired by Tumblin and Turk's work [195].

The first step of the framework is to decompose a HDR image in luminance and chromaticity, for the whole pipeline see Figure 2.24. At this point the luminance in the logarithmic domain is filtered with the bilateral filter, and the detail layer is calculated by dividing the luminance by the filtered one. Then, the filtered luminance is tone mapped using a global TMO. Authors used Tumblin and Rushmeier's operator [194], but any TMO is suitable for the range reduction. Finally, the tone mapped base layer, the detail layer and the chromaticity are recombined to form the final tone mapped image.

Moreover, the authors presented a speed-up of the filtering using an approximated bilateral filtering and down-sampling. However, this technique was made obsolete after the introduction of the bilateral grid which allows real-time performances exploiting graphics hardware [30]. The framework can preserve most of the fine details and can be applied to any global TMO, see Figure 2.25 for an example. The main problem is that halos are not completely removed, in fact they can appear in some images. A further improvement of this general framework was proposed by Choundhury and Tumblin [32], where they employed the trilateral filter for the

(a) (b)

Figure 2.25: *A comparison of tone mapping with and without using the framework proposed by Durand and Dorsey [194] applied to the Mansion HDR image: a) Tumblin and Rushmeier's operator. b) The bilateral framework implemented using the Tumblin and Rushmeier's operator, note that fine details in the sky and in the foliage are enhanced compared to a).*

tone mapping task, Appendix A, see Figure 2.26 for an example.

(a) (b)

Figure 2.26: *A comparison between the fast bilateral filtering [194] and trilateral filtering [31] applied to the Mansion HDR image: a) The image tone mapped with the bilateral filter. b) The image tone mapped with the trilateral filter. Note that details are better reproduced than in a), especially in the sky.*

Kuang et al. [101] modified the Bilateral filter's framework adding features from iCAM 2002 [59], Hunt's model [84], and CIECAM02 [138] for the compression of the base layer. Moreover, a power function is applied to the detail layer to simulate Stevens effect, which predicts the increase of the local contrast when luminance levels are increased. After the combination of the base and detail, an inverse model of characterisation of the displaying device is applied to the image, see Figure 2.27 for an example.


(a) (b)

Figure 2.27: *A comparison between the iCAM 2006 by Kuang et al. [101] and iCAM 2002 by Fairchild and Johnson [59] applied to Nigara Falls HDR image: a) The image processed with iCAM 2002, note that there is a purple colour shift. b) The image processed with iCAM 2006, note that fine details are better preserved than in a) due to the bilateral decomposition. The original HDR image is courtesy of Mark Fairchild [57].*

Gradient Domain Compression

A different approach was proposed by Fattal et al. [61], where gradients of the image are modified to achieve range compression. The main idea is based on the fact that drastic changes in the luminance in an HDR image are due to large magnitude gradients at a certain scale. On the other hand, fine details depend on small magnitude gradients. Therefore, the range can be obtained by attenuating the magnitude of large gradients, while keeping or penalising less small gradients to preserve fine details. This method is inspired by Horn's work on recovering reflectances [81].

In the 1D case this idea is quite straightforward. If a HDR signal $H(x) = \log(L_{\mathrm{w}}(x))$ needs to be compressed, derivatives have to be scaled by a spatially varying factor Φ, which modifies the magnitude, only keeping the direction. This produces a new gradient field $G(x) = H(x)'\Phi(x))$, which is used to reconstruct the compressed signal as:

$$\log(L_{\mathrm{d}}(x)) = I(x) = C + \int_0^x G(t)dt \tag{2.62}$$

where C is an additive constant. Note that computations are performed in the logarithm domain, because it is an approximation of the perceived brightness, and gradients calculated in this domain correspond to ratios, local contrast, in the linear domain. This approach can be extended to the 2D case, for example $G(\mathbf{x}) = \nabla H(\mathbf{x})\Phi(\mathbf{x})$. However, G is not necessarily integrable, because a I may not exist such that $G = \nabla I$. An alternative is to minimise a function I

whose gradients are closest to G:

$$\int\int \left\|\nabla I(x,y) - G(x,y)\right\|^2 dxdy = \tag{2.63}$$

$$= \int\int \left(\frac{\partial I(x,y)}{\partial x} - G_x(x,y)\right)^2 \left(\frac{\partial I(x,y)}{\partial y} - G_y(x,y)\right)^2 dxdy \tag{2.64}$$

which must satisfy the Euler-Lagrange equation, in according with the Variational Principle:

$$2\left(\frac{\partial I(x,y)}{\partial x} - G_x(x,y)\right) + 2\left(\frac{\partial I(x,y)}{\partial y} - G_y(x,y)\right) = 0 \tag{2.65}$$

This equation, once it is simplified and rearranged, leads to Poisson's Equation:

$$\nabla I^2 - \nabla \cdot I = 0 \tag{2.66}$$

which is a linear partial difference equation and can be solved using a full multi-grid method [163]. For Φ, Fattal et al. proposed a multi-resolution definition, because edges are contained at multiple scales. To avoid halos, the attenuation of edges has to be propagated from the level in which it was detected to the full resolution scale. This is achieved by firstly creating a Gaussian pyramid $(H_0, \ldots H_d)$, where H_0 is the full resolution HDR image and H_d is at least a 32×32 HDR image. Secondly, gradients are calculated for each level using central differences:

$$\nabla H_k(x,y) = \left(\frac{H_k(x+1,y) - H_k(x-1,y)}{2^{k+1}}, \frac{H_k(x,y+1) - H_k(x,y-1)}{2^{k+1}}\right) \tag{2.67}$$

At each level k a scaling factor is defined as:

$$\varphi_k(x,y) = \alpha^{1-\beta} \|\nabla H_k(x,y)\|^{\beta-1} \tag{2.68}$$

where α is a constant-threshold, assuming $\beta \leq 1$. While magnitudes above α will be attenuated, ones below α will be slightly magnified. The final Φ is calculated by propagating the scaling factors φ_k from the coarsest level to the full resolution one:

$$\Phi(x,y) = \Phi_0(x,y) \qquad \Phi_i = \begin{cases} \varphi_d(x,y) & \text{if } i = d \\ U\left(\Phi_{k+1}\right)(x,y)\varphi_d(x,y) & \text{if } i < d \end{cases} \tag{2.69}$$

where U is a linear up-sampling operator.


(a) (b) (c)

Figure 2.28: *An example of tone mapping using the gradient domain operator by Fattal et al. [61] varying β parameter applied to Stanford Memorial Church HDR image: a) $\beta = 0.7$. b) $\beta = 0.8$ c) $\beta = 0.9$. Note that increasing β the overall look of the image gets darker. The original HDR image is courtesy of Paul Debevec [47].*

This operator, preserves fine details avoiding halo artifacts by being based on differential analysis, see Figure 2.28. Moreover, the operator is a very general solution, and it can be employed for the enhancement of LDR images preserving shadow areas and details.

2.4.4 Segmentation Operator

Recently, a new approach to the tone mapping problem has emerged with segmentation operators. Strong edges and most of local contrast perception are located between the border of big uniform regions. The idea is to divide the image into uniform segments, to apply a global operator to each segment, and finally to merge them. Furthermore, an additional advantage is that gamut modifications are minimised because a linear operator for each segment sometimes is enough.

Segmentation and Adaptive Assimilation for Detail-Preserving

The first segmentation based TMO was introduced by Yee and Pattanaik [224]. Their operator divides the HDR image into regions, and calculates an adaptation luminance for each one. This adaptation luminance can be used by any global operator as a local input.


(a) (b)

Figure 2.29: *An example of Tumblin Rushmeier's TMO [193] using adaptive luminance calculated by the segmentation method by Yee and Pattanaik [224] applied to the Bottles HDR image: a) The adaptation luminance is calculated with only 5 layers, in this case bands are visible creating artifacts. b) The adaptation luminance is calculated with 32 layers, in this case bands are not visible.*

The first step of the segmentation is to divide the image into regions, called categories, using a histogram in the logarithm domain. Contiguous pixels of the same category are grouped together using a flood-fill approach. Finally, small groups are assimilated by a bigger one, obtaining a layer. Small and big groups are defined by two thresholds, tr_{big} and tr_{small}. The segmentation is performed again for n_{max} layers, where the difference is in the size of the bin of the histogram, which is computed as:

$$Bin_{size}(n) = Bin_{min} + (Bin_{max} - Bin_{min})\frac{n}{n_{max} - 1} \tag{2.70}$$

where $Bin_{size}(n)$ is the bin size for the n-th layer, and Bin_{max} and Bin_{min} are respectively the minimum and maximum bin size. Once all layers are computed, the adaptation luminance is calculated as:

$$L_{\mathrm{a}}(\mathbf{x}) = \exp\left(\frac{1}{n_{max}}\sum_{i=1}^{n_{max}} C_i(\mathbf{x})\right) \tag{2.71}$$

where $C_i(\mathbf{x})$ is the average log luminance of the group which pixel at coordinate $\mathbf{x}$ belongs to. The application of L_{a} to any global TMO helps to preserve edges avoiding halos. However, the algorithm needs a high number of layers to avoid banding artifacts, considerably increasing the computational time, see Figure 2.29.

Lightness Perception in Tone Reproduction

Krawczyk et al. [99] proposed an operator based on the anchoring theory of the lightness perception by Gilchrist et al. [68]. This theory states that the highest luminance value, or anchor, in the visual field is perceived as white by the HVS. The perception is affected by relative area, when the highest luminance covers a small area it appears to be self luminous. To apply lightness theory to complex images, Gilchrist et al. [68] proposed to decompose the image in areas, called frameworks, where the anchoring can be applied.

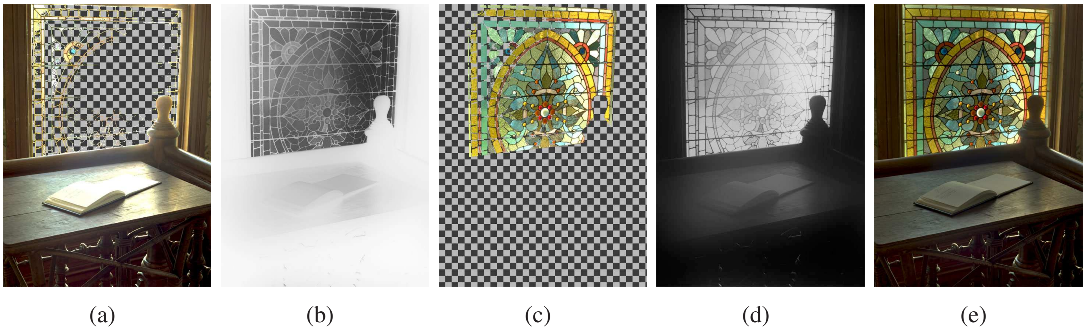

(a) (b) (c) (d) (e)

Figure 2.30: *An example of TMO by Krawczyk et al. [99] applied to the Desk HDR image: a) The first framework where anchoring is applied. b) The smoothed probability map for the first framework. c) The second framework where anchoring is applied. d) The smoothed probability map for the second framework. e) The final tone mapped image is obtained by merging the first framework b) and the second one d). Images are courtesy of Karol Myszkowski [99].*

The first step of the operator is to determine the frameworks. Afterwards, the image's histogram is calculated in the $\log_{10}$ domain. K-means clustering algorithm is used to determine centroids, C_i, in the histogram, merging close centroids by a weight averaging based on pixel count. To avoid seams or discontinuity, frameworks are generated with a soft segmentation. A probability function is defined for each framework, which states if a pixel belongs to the framework based on the centroid:

$$P_i(\mathbf{x}) = \exp\left(-\frac{\left(C_i - \log_{10}(L_{\mathrm{w}}(\mathbf{x}))\right)^2}{2\sigma^2}\right) \tag{2.72}$$

where σ is equal to the maximum distance between two frameworks. $P_i(\mathbf{x})$ is smoothed using the bilateral filter to remove small local variations, see Figure 2.30.b, Figure 2.30.d. The local anchor, ω_i, for a framework, i, is calculated by calculating the 95-th percentile of luminance in the framework. Finally, the tone mapped image is calculated as:

$$L_{\mathrm{d}}(\mathbf{x}) = \log_{10}(L_{\mathrm{w}}(\mathbf{x})) - \sum_{i=1}^{n} \omega_i P_i(\mathbf{x}) \tag{2.73}$$

An example of the final tone mapped image can be seen in Figure 2.30. The operator was validated comparing it with the photographic tone reproduction [169] and the fast bilateral filtering operator [55] to test the Gelb effect [68], an illusion related to lightness constancy failure. The result of the experiment showed that the lightness based operator can reproduce this effect. The operator is fast and straightforward to implement, but it needs particular care when applying it to dynamic scenes to avoid effects such as ghosting.

Interactive Local Manipulation of Tonal Values

An user based system for modifying tonal values in a HDR/LDR image was presented by Lischinski et al. [118]. The system is based on a brush interface inspired by Levin et al. [112] and Agarwala et al. [5].

The user specifies using brushes which regions in the image require an exposure adjustment. There are four possible brushes available :

- **Basic brush**: sets constraints to pixels covered by the brush, assigning a weight of $w = 1$.
- **Luminance brush**: applies a constraint to pixels whose luminance is similar to those covered by the brush. If μ is the mean luminance for pixels under the brush, and σ is a brush parameter, a pixel with luminance L is assigned a weight of $w(l) = \exp\big(-(L-\mu)^2\sigma^{-2}\big)$.
- **Luma-chrome brush**: is a brush similar to the Luminance one but it takes into account luminance and chromaticity.
- **Over-exposure brush**: selects in a given all over-exposed pixels area which are surrounded by the drawn stroke.

Once the stroke is drawn using the various brushes, the system tries to find a spatially varying exposure function f. This has to satisfy the constraints specified by the user and to take into account edges. In terms of minimisation, f can be defined as:

$$f = \operatorname{argmin}_f \left(\sum_{\mathbf{x}} w(\mathbf{x}) \big(f(\mathbf{x}) - g(\mathbf{x}) \big)^2 + \lambda \sum_{\mathbf{x}} h(\nabla f, \nabla L') \right) \tag{2.74}$$

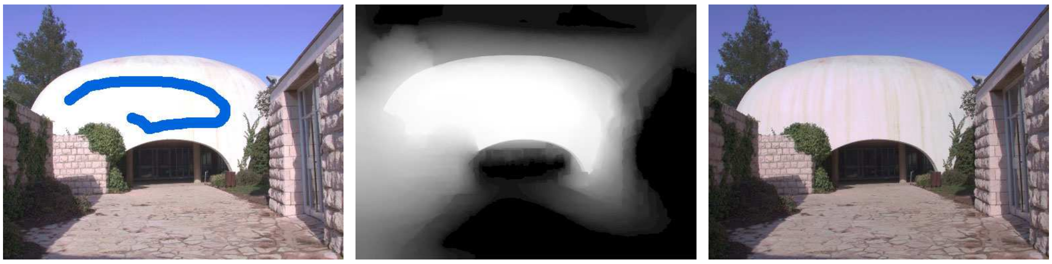

(a) (b) (c)

Figure 2.31: *An example of the user based system by Lischinski et al. [118] applied to the Synagogue HDR image: a) The user decided to reduce the exposure in the over exposed dome, so a stroke is drawn in that area. b) At this point the linear system solver generates a smooth exposure field keeping strong edges. c) The final result of the adjustment of the stroke in a). Images are courtesy of Daniel Lischinski [118].*

where $L' = \log(L)$ and L is the luminance of a LDR or HDR image. $w(\mathbf{x}) \in [0,1]$ is a function that defines constrained pixels, $g(\mathbf{x})$ is a function that defines the target exposure. The minimisation presents a smoothing term, that takes into account large gradients, and is defined as:

$$h(\nabla f, \nabla L') = \frac{|f_x|^2}{|L'_x|^\alpha + \varepsilon} + \frac{|f_y|^2}{|L'_y|^\alpha + \varepsilon} \tag{2.75}$$

where α defines the sensitivity of the term to the derivatives of the log-luminance image, ε is a small non zero value to avoid singularities. The minimum f is calculated by solving a linear system [163]. Afterwards, the image is adjusted applying f, see Figure 2.31.


(a) (b) (c)

Figure 2.32: *An example of the automatic operator by Lischinski et al. [118] applied to the Belgium HDR image: a) The exposure map for each zone. b) The smoothed exposure map, note that sharp edges are kept. c) The final tone mapped image. Images are courtesy of Daniel Lischinski [118].*

Finally, the system presents an automatic TMO for preview which follows the Zone System by Adams [1]. The image is segmented in $n = \lceil \log_2(L_{\text{Max}}) - \log_2(L_{\text{Min}} + 10^{-6}) \rceil$ zones, and the correct exposure is calculated for each zone. These exposures and segments are used as an input for the linear system solver, generating the tone mapped image, see Figure 2.32 for an example.

The system presents an user friendly GUI and a complete tool for photographer, artists and final users to tone intuitively map HDR images, or enhance LDR images. Moreover, it can be applied to other tasks such as modification of the depth of field and white balance.

Exposure Fusion

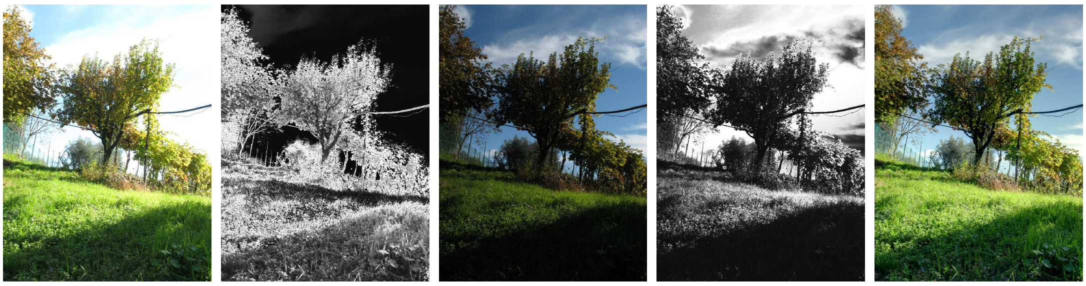
HDR images usually need to be assembled from a series of LDR ones before tone mapping. A novel approach that can avoid this step was proposed by Mertens et al. [131]. The idea of the operator is to merge well-exposed pixels from each exposure.

The first step is the analysis of each LDR image to determine which pixels can be used during merging. This is achieved calculating three metrics for each pixel: contrast, saturation, and well-exposedness. Contrast, C, refers to the absolute value of the gradients in the image. Saturation, S, is defined as the standard deviation of the red, green and blue channels. Well-exposedness, E, of luminance L determines if a pixel is well-exposed in a fuzzy way as $E(L) = \exp(-0.5(L-0.5)^2(\sigma)^{-2})$. These three metrics are combined together, obtaining a weight $W_i(\mathbf{x})$ which determines the importance of the pixel for that exposure:

$$W_i(\mathbf{x}) = C_i(\mathbf{x})^{\omega_C} \times S_i(\mathbf{x})^{\omega_S} \times E_i(\mathbf{x})^{\omega_E} \tag{2.76}$$

where i refers to the i-th exposure, and ω_C, ω_S, and ω_E are exponents to increase the influence of a metric over the others. Weights are normalised by $T(\mathbf{x}) = \sum_{i=1}^{n} W_i(\mathbf{x})$ to obtain consistent results.

After analysis, the exposures are combined in the final image. To avoid seams and discontinuity the images are blended using Laplacian pyramids [24]. While the weights are decomposed in Gaussian pyramids denoted with the operator $\mathbf{G}$, exposure images, I_k are decomposed in Laplacian pyramids denoted with the operator $\mathbf{L}$. Therefore, the blending is calculated as:



(a) (b) (c) (d) (e)

Figure 2.33: *An example of the fusion operator by Mertens et al. [131] applied to the Tree HDR image: a) The first exposure of the HDR image. b) The weight map for a), note that pixels from the tree and the ground have high weights because they are well-exposed. c) The second exposure of the HDR image. d) The weight map for c), note that pixels from the sky have high weights because they are well-exposed. e) The fused / tone mapped image using Laplacian pyramids.*

$$\mathbf{L}^l\{I_\mathrm{d}\}(\mathbf{x}) = \sum_{i=1}^{n} \mathbf{L}^l\{I_i\}(\mathbf{x})\mathbf{G}^l\{W_i\}(\mathbf{x}) \tag{2.77}$$

where l is the l-th level of the Laplacian/Gaussian pyramid. Finally, the $\mathbf{L}\{I_\mathrm{d}\}$ is collapsed obtaining the final tone mapped image I_d, see Figure 2.33.

The main advantage of this operator is that a consumer user does not need to generate HDR images, also it minimises colour shifts that can happen in traditional TMOs. This is because well-exposed pixel are taken in the blend without applying a real compression function, but just a linear scale.

2.5 Native Visualisation of HDR Content

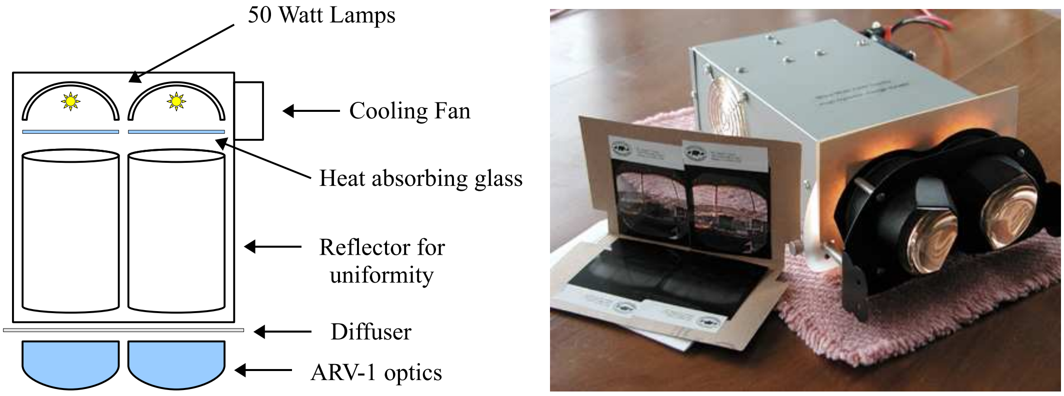
Only in the last few years, researchers have been working on display technologies with the aim of native visualisation of HDR images and videos without using TMOs. The two main devices are the HDR viewer [211, 179] and the HDR Monitor [179], for an overview on these device see Table 2.5. Both of them divide an HDR image into a detail layer with colours and a luminance layer that back-modulates the first one.

Device	L_{Max} in cd/m^2	L_{Min} cd/m^2	Dynamic Range
HDR Viewer	5,000	0.5	10,000:1
HDR Monitor Projector based	2,700	0.054	50,000:1
HDR Monitor LED based 37”	3,000	0.015	200,000:1

Table 2.5: *A summary of the built HDR devices and their main features.*

2.5.1 The HDR Viewer

Ward [211] and Ledda et al. [110] presented the first native viewer of HDR images, see Figure 2.34. Their device is inspired by the classic stereoscope, a device used at the end/beginning of the 19th/20th century for displaying 3D images.



(a) (b)

Figure 2.34: *The HDR viewer by Ward [211] and Ledda et al. [110]: a) A scheme of the HDR viewer. b) A photograph of the HDR viewer prototype. The photograph is courtesy of Greg Ward [211].*

The HDR viewer is composed by three main parts: two lenses, two 50 Watt lamps, and two film transparencies, one for each eye, that encodes an image taken/calculated at a slightly different camera position to simulate the depth effect. The two lenses are large expanse extra perspective (LEEP) ARV-1 optics by Erik Howlett [83], which allow a 120 degree field of view. Moreover, an extra transparency image is needed for each eye that increases the dynamic range through the light source modulation, because a film transparency can encode only 8-bit images due to limits of the medium. Note that, when light passes through a transparent surface it is modulated, a simple multiplication, by the level of transparency.

The processing method splits an HDR image in two, for the complete pipeline see Figure 2.35. The first one, that is used to modulate the light source, is created by applying a 32×32 Gaussian filter to the square root of the image luminance. The second image, in front of the one for modulation, is generated by dividing the HDR image by the modulation one. To take into account the chromatic aberration of the optics, the red channel is scaled 1.5% more than blue one, with the green channel halfway between. Note that while the image in front encodes colours and details, the back one for modulation encodes the global luminance distribution.

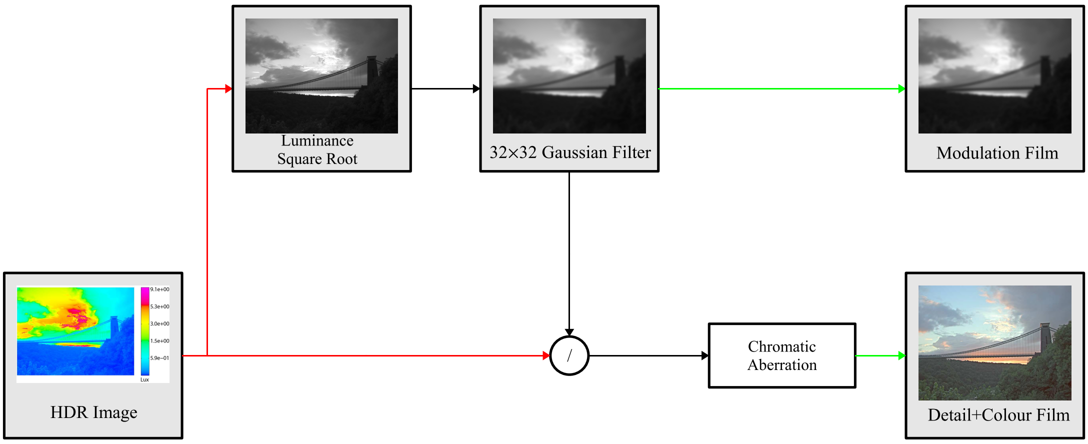


Figure 2.35: *The processing pipeline to generate two images for the HDR viewer by Ward [211] and Ledda et al. [110].*

The device and the processing technique allow to display images with a 10,000:1 dynamic range, where the measured maximum and minimum luminance are respectively 5,000 cd/m^2 and 0.5 cd/m^2. Ledda et al. [110] validated the device against reality and the histogram adjustment operator [107] on a CRT monitor using a series of psychophysical experiments. The results showed that the HDR viewer is closer to reality than a TMO reproduced on a CRT monitor.

The system was the first solution for the native visualisation of HDR images. However, it is limited to display only static images, and the cost of printing the four film transparencies is around $200 U.S. Finally, only one person can use it at a time.

2.5.2 HDR Monitors

Seetzen et al. [179] developed the first HDR Monitors based on two technologies: the DLP projector, and light emitting diode (LED). As for the HDR viewer, there is a modulated light source which boosts the dynamic range of a front layer that encodes details and colours. Both technologies are based on LCD monitors for visualising the front layer.

The first developed technology is the DLP projector one, where a projector is driven to modulate the light, see Figure 2.36. The processing method for creating images for the LCD and the projector is similar to the method for the HDR viewer in Section 2.5.1. However, there are few differences. Firstly, chromatic aberration correction is removed because there is no optics.

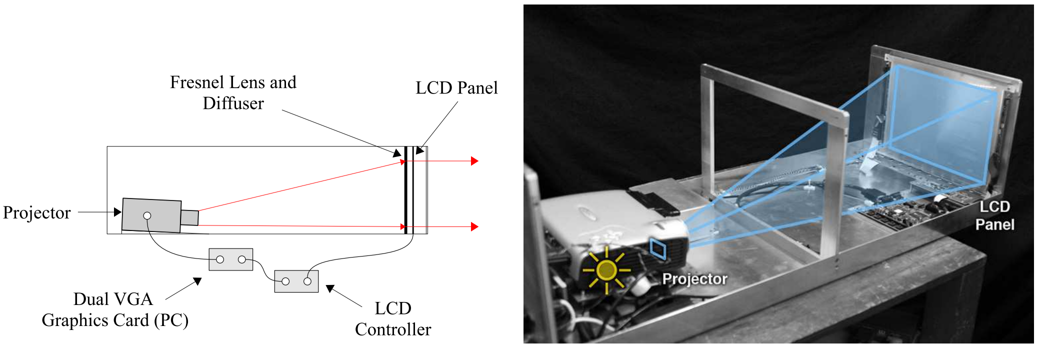
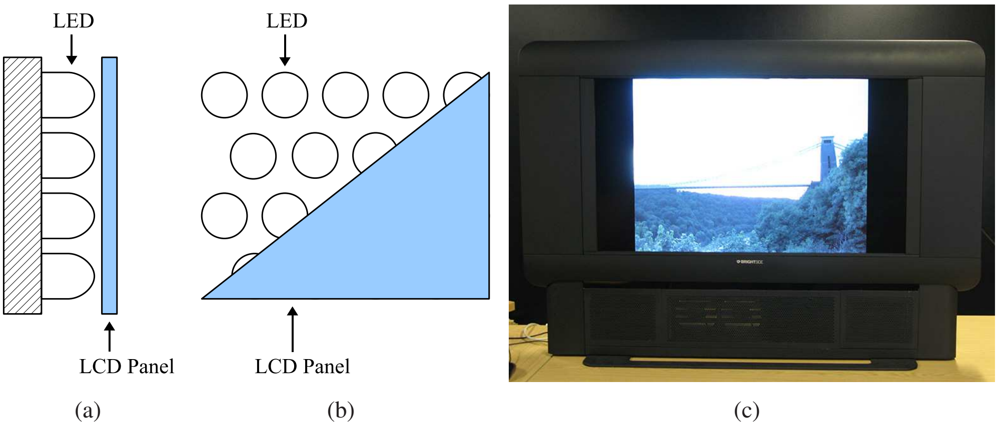


Figure 2.36: *The HDR Monitor based on projector technology: a) A scheme of the monitor b) A photograph of the HDR Monitor. The photograph is courtesy of Matthew Trentacoste [179].*

Secondly, the filtering of the square root luminance is modelled on the point spread function of the projector. Finally, the response functions are measured for both LCD panel and projector, and their inverses are applied to the modulation image and front image for linearising the signal.

The measured dynamic range of the 15.1” prototype monitor is 50,000:1 dynamic range, where the measured maximum luminance level is 2,700 cd/m^2 and the measured minimum luminance level is 0.054 cd/m^2. For this monitor the LCD panel is a 15.1” Sharp LQ150X1DG0 with a dynamic range of 300:1, and the projector is a Optoma DLP EzPro737 with a contrast ratio of 800:1. Nevertheless, this technology has a series of problems. The required optical path by the projector is large, around one meter for a 15.1” monitor, which is not practical for home entertainment and bigger displays. Moreover, the viewing angle is very small, because a Fresnel lens, which is used for uniforming luminance values, has a huge fall-off at wide viewing angles. Finally, the projector needs to be very bright, and this causes high power consumption and heat.

The second technology is the LED based one, where a low resolution LED monitor is driven to modulate the light, see Figure 2.37. The processing algorithm for the generation of images for the LED monitor and the LCD panel is similar to the one for the HDR Monitor based on projectors [192]. The main difference is the addition of a step in which the luminance for each LED is determined based on a down-sampled square root luminance to the resolution of the LED monitor and solving for the values, taking the overlap of the point spread function of the LED into account. The main LED model is the DR-37P, a 37” HDR Monitor with a 200,000:1



(a) (b) (c)

Figure 2.37: *The HDR Monitor based on LCD and LED technologies: a) The scheme of a part of the monitor in a lateral section. b) The scheme of a part of the monitor in a frontal section c) A photograph of the HDR Monitor.*

dynamic range where the measured maximum and minimum luminance level are respectively 3,000 cd/m^2 and 0.015 cd/m^2. For this monitor, 1,380 Seoul Semiconductor PN-W10290 LEDs were mounted behind a 37” Chi Mei Optoelectronics V370H1L01 LCD panel with a contrast ratio of 250:1 [49].

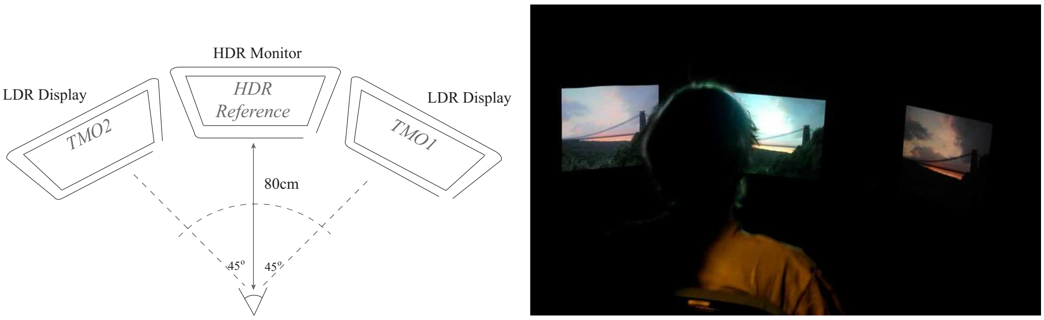
However, even this display technology has some problems. The first one is the quality. While the dynamic range is increased, the image quality is reduced having a lower back-modulated resolution than the projector technology. The second one is the LEDs which are very expensive, consume a lot of power (1,680 W), and produce heat. The heat dissipation is carried out using fans and a liquid base system, but results in a reasonably loud system.

2.6 Evaluation of Tone Mapping Operators

In the last few years, researchers have started to investigate the performances of TMOs. The large number of TMOs created the need to measure the quality of a TMO for a given task, or on overall as a reproduction of the real world. These studies highlighted that the best TMO has not yet been found, but they found some correlation between attributes of an image. In this section the main studies regarding tone mapping evaluation are presented.

2.6.1 Psychophysical Experiments

The main tool for measuring performances of a TMO is to run psychophysical experiments rather than using metrics, because there is no metric that can fully simulate the HVS in all its complexity for HDR values. For example, HDR-VDP [127, 124], which is considered the de facto standard in HDR imaging, works only on the luminance channel without using any colour information as it provides important stimuli in human vision.



(a) (b)

Figure 2.38: *An example of the setup for the evaluation of TMOs using an HDR monitor as reference: a) The diagram. b) A photograph. The photograph is courtesy of Patrick Ledda [108].*

The setup of a psychophysical experiment for tone mapping consists of a dark room, to avoid interference's by other stimuli, and calibrated HDR and/or LDR displays. While LDR monitors are used to visualise tone mapped images, HDR monitors are generally employed as HDR references, see Figure 2.38. Nevertheless, a real scene can be used as well. A subject of an experiment is a person with normal or corrected to normal vision who is instructed about the task that has to be performed. He/She usually performs the task interacting with a computer interface that collects data. There are three main kinds of psychophysical experiments that have been used in the evaluation of TMOs:

- **Ranking**: each subject has to rank a series of stimuli based on a criteria. In this case results are more accurate and precise because a human being has to make a firm decision. The main disadvantage is that the task is time consuming because a subject has to compare all images. Ranking can be performed indirectly using pairwise comparisons.
- **Rating**: a subject has to rate on a scale an attribute of a stimulus. A reference or another stimulus can be included. This method is very fast to perform because all other stimuli

are not needed. However, subjects can have different perceptions of the rating scale so collected data can be less precise than ranking experiments.

- **Pairwise Comparisons**: a participant needs to determine which image in a pair is closer to a reference or which image in a pair satisfies better a certain property. The method produces data with less subjective problems, such as rating, and there are statistics to determine the significance of inferred ranks unlike ranking. Nevertheless, the method usually requires more time than rating and ranking. Each subject needs to evaluate $n(t-1)t/2$ pairs where t is the number of TMOs and n is the number of images.

2.6.2 Perceptual Evaluation of Tone Mapping Operators with Regard to Similarity and Preference

One of the first studies in computer graphics on tone mapping evaluation was conducted by Drago et al. [51], where the main goal was to measure the performances of TMOs applied to different scenes. The study consisted of two parts. The first one is a pyschophysical experiment in which participants answered subjective questions. The second one is a multi-dimensional statistical analysis inspired by Pellacini et al.'s work [158]. Seven TMOs were tested: LCIS [195], revised Tumblin-Rushmeier [193], photographic tone reproduction [169], uniform rational quantisation [178], histogram adjustment [107], Retinex [165], and visual adaptation [63].

The results of the analysis showed that the photographic tone reproduction operator tone mapped images closest to the ideal point extracted from preferences of participants. Moreover, this operator, the uniform rational quantisation, and Retinex methods were in the same group for better looking images. This is probably due to the their global contrast reduction operators which shares many common aspects. While visual adaptation and revised Tumblin-Rushmeier were in the second group, the histogram adjustment was between the two groups.

The study showed a methodology for measuring the performance of a TMO using subjective data. However, the main problem is the number of people who took part, 11 subjects, and the size of the data-set, 4 images, which are too small for drawing significance conclusions. Drago et al. [50] used these findings to design a Retinex TMO, which was not counter-evaluated to check if the desired quality was reached.

2.6.3 Evaluating HDR Rendering Algorithms

Kuang et al. made two studies [102, 103] on the evaluation of TMOs. The first one [102] showed the correlation in ranking between coloured and grey scale images. More interesting is the second study which extended the first one by introducing a methodology for testing TMOs for overall preference and accuracy. This study is divided in three experiments where 33 participants took part in them. The first experiments had the goal to determine participants preferences amongst TMOs. The second one was designed to measure attributes that help to choose a TMO among others. Finally, the third experiment had the goal to measure the accuracy for a TMO of reproducing real world scenes. Six TMOs were tested: histogram adjustment [107], Braun and Fairchild's sigmoid transform for colour gamut [23], Funt et al.'s Retinex [67], iCAM 2002 [59], a modified fast bilateral filtering [55], and photographic tone reproduction [169].

The results of the first experiments showed that coloured and grey scale images were correlated in the ranking. The second experiment highlighted that overall preference was correlated with details in dark areas, overall contrast, sharpness and colourfulness. Moreover, the overall appearance can be predicted by a single attribute. Finally, the third one showed that data of the first part (preferences) and one of the second part (accuracy with the real world) were correlated. Therefore, the authors suggested that both methodologies can be utilised as an evaluation paradigm.

In all experiments the modified fast bilateral filtering outperformed the task, suggesting that it is a good algorithm for tone mapping.

2.6.4 Paired Comparisons of Tone Mapping Operators using an HDR Monitor

One of the first studies using an HDR reference was presented by Ledda et al. [108], where an HDR monitor was employed as the ground truth. This study used a pairwise comparison methodology with a reference [43]. 48 subjects took part in the experiments, and 23 images were tone mapped using 6 TMOs: histogram adjustment operator [107], fast bilateral filtering operator [55], photographic tone reproduction operator [169], iCAM 2002 [59], adaptive logarithmic mapping [52], and local eye adaptation operator [109].

The collected data was analysed using accumulated pairwise preference scores [43] in combination with coefficients of agreement and consistency [93]. The study showed that on overall the photographic tone reproduction and iCAM 2002 performed better than the other operators, but in the case of grey scale images the first operator was superior. This is due to the fact the iCAM processes primarily colours, and it is disadvantaged in a grey scale context. Moreover, iCAM 2002 and the local eye adaptation operator performed very well for detail reproduction in bright areas. In conclusion, iCAM 2002 performed generally better than other operators, because colours are an important stimuli in human vision. Furthermore, the study highlighted low performances of the fast bilateral filtering operator, which can be due to the fact that high frequencies are exaggerated, when compared to the reference.

This study showed a robust methodology, where a large number of participants took part in those experiments, and a large data set was used covering all varieties of scene categories such as: day, night, outdoor, indoor, synthesised images, and captured images from the real world.

2.6.5 Testing TMOs with Human-Perceived Reality

A different approach to the evaluation was proposed by Yoshida et al. [225, 226], who ran a series of subjective psychophysical experiments using, as reference, a real world scene. The main goal of the study was to assess the differences in the perception of different TMOs against the real world. This was achieved by measuring some attributes such as image naturalness, overall contrast and brightness, and detail reproduction in dark and bright areas of the image. 14 participants took part in this experiment, and 7 TMOs were tested: linear mapping, histogram adjustment [107], time dependent visual adaptation [156], Ashikhmin operator [10], fast bilateral filtering [55], photographic tone reproduction [169], and adaptive logarithmic mapping [52].

The first finding is that there is no statistical difference between the two scenes. The second one is that the perception of naturalness seems to have no relationship with other attributes. While global TMOs performed better than local ones in terms of overall brightness and contrast, local TMOs performed better for detail reproduction in bright areas. However, Ashikmin's operator and adaptive logarithmic mapping performed better than others for detail reproduction in dark areas. Finally, the most natural operators were the photographic tone reproduction, histogram adjustment and adaptive logarithm. The authors also compared these results with Ledda et al.'s study [108] and Kuang et al.'s one [102, 103] noticing that their study shares some similar results with this previous work.

This study analysed five attributes of an image. The provided analysis determined which TMO can perform better than others for a given task, such as reproduction of contrast, brightness, reproduction of detail in bright and dark regions, and naturalness. Nevertheless, the main problem of this work is that only two indoor scenes with artificial illumination were used in the experiment.

2.6.6 Image Attributes and Quality for Evaluation of Tone Mapping Operators

Čadík et al. [25, 26] presented a TMO evaluation study with subjective psychophysical experiments using as reference real scenes as in Yoshida et al. [225, 226]. Both rating and ranking methodologies were employed. Furthermore, collected data was fitted into different metrics. The main focus of the study was on some image and HVS attributes such as: brightness or perceived luminance, contrast, reproduction of colours, and detail reproduction bright and dark areas of the image. 10 participants took part in the two experiments, and 14 TMOs where employed: linear mapping, LCIS [195], revised Tumblin-Rushmeier [193], photographic tone reproduction [169], uniform rational quantisation [178], histogram adjustment [107], fast bilateral filtering [55] , trilateral filter [32], adaptive gain control [154], Ashikhmin operator [10], gradient domain compression [61], contrast based factor [210], adaptive logarithmic mapping [52], and spatially non-uniform scaling [31].

The first finding is there is no statistically difference between the data of the two experiments, rating and ranking. Therefore, the authors suggested that for a perceptual comparison of TMOs ranking without a reference is enough. The second one is the good performance of global methods over local ones. This fact was confirmed with other studies such Ledda et al. [108], Akyüz et al. [7], Drago et al. [51], and Yoshida et al. [225, 226]. In the last part of the study, the relationship between the overall image quality and the four attributes was analysed, and fitted into parametric models for generating image metrics.

The study measured performances of a large number of TMOs. Furthermore, four important attributes of an image were measured, and not only the overall quality. However, the number of scenes is very limited as in the case of Yoshida et al. [225, 226], not covering the other common real world lighting conditions.

2.6.7 A Reality Check for Tone-Mapping Operators

Another study on evaluation of TMO is based on ranking with real scenes as references was proposed by Ashikhmin and Goyal [11]. The study explored how people perceive real scenes when compared to TMOs, realism of TMOs and personal preference of them. 15 subjects took part in the experiments, and 5 TMOs were tested: histogram adjustment [107], gradient domain compression [61], photographic tone reproduction [169], adaptive logarithmic mapping [52], and trilateral filtering [32].

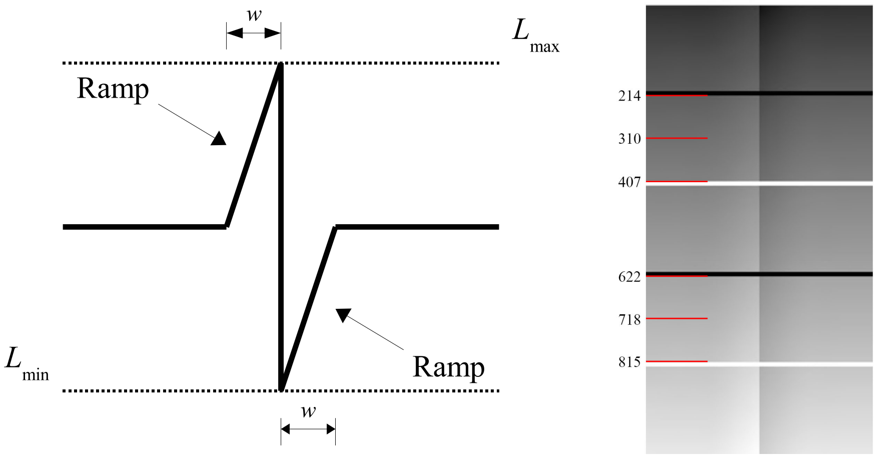
The study showed that the difference between experiments with and without the reference is large, in disagreement with Čadík et al. [25, 26]. The main problem of the proposed study, as other ones which use a real scene as a reference, is the limited number of analysed scenes, which are mostly indoor with artificial illumination. This is due to the problem of controlling the scene in outdoor environment, but it does not allow the study to be generalised to all lighting conditions and scenes. Furthermore, the first experiment was asking a personal preference, which depends greatly for each subject.

2.6.8 Perceptual Evaluation of Tone Mapping Operators using the Cornsweet-Craik-O'Brien Illusion

Akyüz and Reinhard [9] made a study in which contrast in tone mapped images was isolated and evaluated for a better understanding of the attribute. This was achieved using synthesised stimuli, in particular they chose to evaluate the Cornsweet-Craik-O'Brien illusion [35]. This is characterised by a ramp, Figure 2.39.a, between two flat regions that increases the perceived contrast as:

$$C = \frac{L_{\max} - L_{\min}}{L_{\max} + L_{\min}} \tag{2.78}$$

where $L_{\max}$ and $L_{\min}$ are respectively the maximum and minimum luminance value of the ramp.



(a) (b)

Figure 2.39: *The Cornsweet-Craik-O'brien illusion used in Akyüz and Reinhard's study [9]: a) The profile needed to generate the illusion. b) The stimulus image used in the experiment. The red lines indicates the scanlines evaluated in the experiment. The stimulus image is courtesy of Ahmet Oğuz Akyüz [9].*

13 subjects took part in the experiment and 7 TMOs were tested: histogram adjustment [107], revised Tumblin-Rushmeier [193], gradient domain compression [61], fast bilateral filtering [55], photographic tone reproduction [169], and iCAM 2002 [59].

The results of the analysis showed that frequency based operator, such as the gradient domain compression and the fast bilateral filtering, did not perform well. This is due to the processing nature of TMOs that cannot handle smooth gradients well. iCAM 2002 performed as the best TMO without statistical difference from the original image. This is due to clipping to dark pixels, that increases contrast despite sigmoid compression. Note that photographic tone reproduction did not perform well in the last two scanlines, because its sigmoid shape compresses the dynamic range too much.

A new methodology of comparisons was presented without the need of a true HDR reference, only a slice of information is judged at a time, in fact each scanline was LDR. The study focused on contrast reproduction assessing that TMOs do not preserve the Cornsweet illusion in the same way. While some TMOs decrease the illusion because gradients are attenuated, others exaggerate it by pronouncing them.

2.7 Image Based Lighting

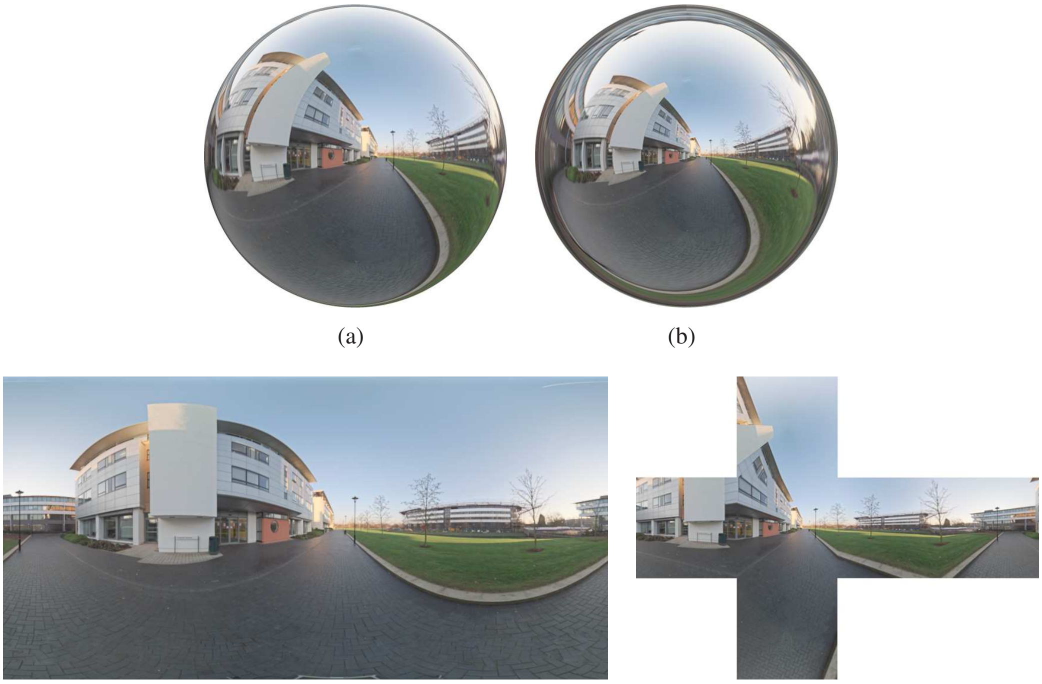
HDR images can simplify the definition and rendering process. In particular, a series of techniques commonly referred to as *Image Based Lighting* are aimed at simulating light transport, defining light sources and the surrounding environment.

2.7.1 Environment Map

IBL usually takes as an input an image, called environment map, that captures irradiance values of the real world environment for each direction, $\mathbf{D} = [x, y, z]^\top$, around a point. Therefore, an environment map can be parameterised on a sphere. Different 2D projection mappings of a sphere can be adopted to encode the environment map. The more popular ones in computer graphics are: spherical mapping Figure 2.40.a, angular mapping Figure 2.40.b, latitude-longitude Figure 2.40.c, and cube-map Figure 2.40.d. See Reinhard et al. [170] or Akenine-Moller and Haines [6] for more details about the mapping methods.

2.7.2 Rendering with IBL

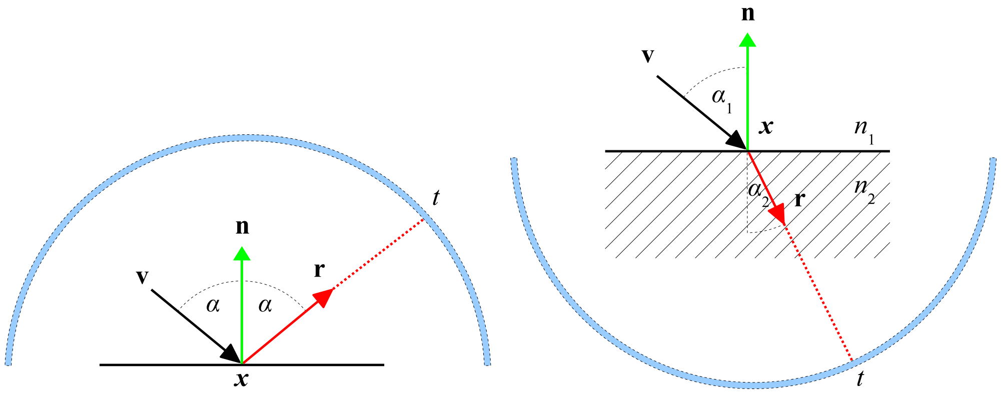
IBL was used to simulate perfect specular effects such as reflection and refraction in the seminal work by Blinn and Newell [21]. The reflected/refracted vector at a surface point $\mathbf{x}$ is used to look-up in the environment map, and the colour at that address is used as reflected/refracted value, see Figure 2.41. This method allows very fast reflection/refraction, see Figure 2.44.a and Figure 2.44.b. However, there are some drawbacks. Firstly, concave objects cannot have internal inter-reflections/refractions, because the environment map does not take into account local features, see Figure 2.42.a. Secondly, reflection/refraction can be distorted since there is a parallax between the evaluation point and the point where the environment map was captured, see Figure 2.42.b.



(a) (b)

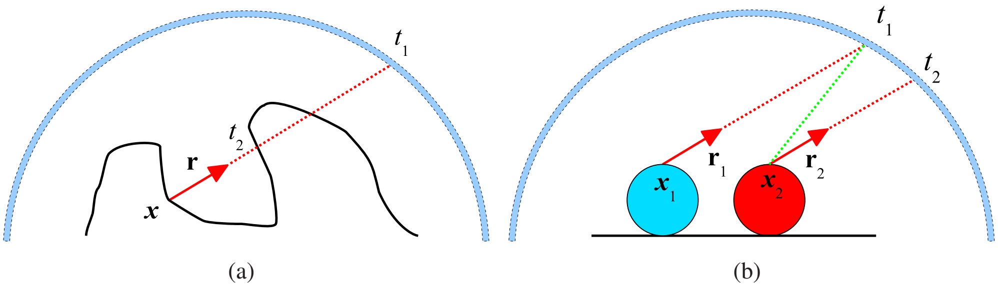
(c) (d)

Figure 2.40: *The Computer Science environment map encoded using the projection mappings: a) Mirrored spherical map. b) Angular map. c) Latitude-longitude map. d) Cube map unfolded into a horizontal cross.*



(a) (b)

Figure 2.41: *The basic Blinn and Newell [21] method for IBL: a) The reflective case, the view vector* $\mathbf{v}$ *is reflected around normal* $\mathbf{n}$ *obtaining vector* $\mathbf{r} = \mathbf{v} - 2(\mathbf{n} \cdot \mathbf{v})\mathbf{n}$*, which is used to look-up into the environment map and get the colour value* t*. b) The refractive case, the view vector* $\mathbf{v}$ *coming from a medium with index of refraction* n_1 *enters in a medium with index of refraction* $n_2 < n_1$*. Therefore,* $\mathbf{v}$ *is refracted following the Snell law,* $n_1 \sin\alpha_1 = n_2 \sin\alpha_2$*, obtaining* $\mathbf{r}$*. This vector is used to look-up into the environment map and get the colour value* t*.*



(a) (b)

Figure 2.42: *The basic Blinn and Newell [21] method for IBL: a) The point* $\mathbf{x}$ *inside the concavity, erroneously uses* t_1 *instead of* t_2 *as colour for refraction/reflection. This is due to the fact the environment map does not capture local features. b) In this case, both reflected/refracted rays for the blue and red objects are pointing to the same direction, but from different starting points. However, the evaluation does not take into account the parallax, so* $\mathbf{x}_1$ *and* $\mathbf{x}_2$ *shares the same colour* t_1.

In parallel Miller and Hoffman [136] and Green [71] extended IBL for simulating diffuse effects, see Figure 2.44.c. This was achieved by convolving the environment map with a low-pass kernel:

$$E(\mathbf{n}) = \int_{\Omega(\mathbf{n})} L(\omega)(\mathbf{n}\cdot\omega)d\omega \tag{2.79}$$

(a) (b)

Figure 2.43: *The Computer Science environment map filtered for simulating diffuse reflections: a) The original environmental map. b) The convolved environment map using Equation 2.79.*

where L is the environment map, $\mathbf{n}$ is a direction in the environment map, $\Omega(\mathbf{n})$ is the positive hemisphere of $\mathbf{n}$. For an example of a convolved environment map see Figure 2.43. In this case, the look-up vector for a point is the normal. Nevertheless, this extension inherits the same problems of the Blinn and Newell's one: no inter-reflection and parallax issues.

Debevec [44] proposed a general method for IBL which takes into account arbitrary BRDF and inter-reflections. In addition he used HDR environment maps, which encode real-world irradiance data. The proposed method is based on ray tracing, and the evaluation, for each pixel, is divided into the following cases:

(a) (b) (c)

Figure 2.44: *An example of classic IBL using environment maps applied to the Stanford's Happy Buddha model [73]: a) Simulation of a reflective material. b) Simulation of a refractive material. c) Simulation of a diffuse material.*

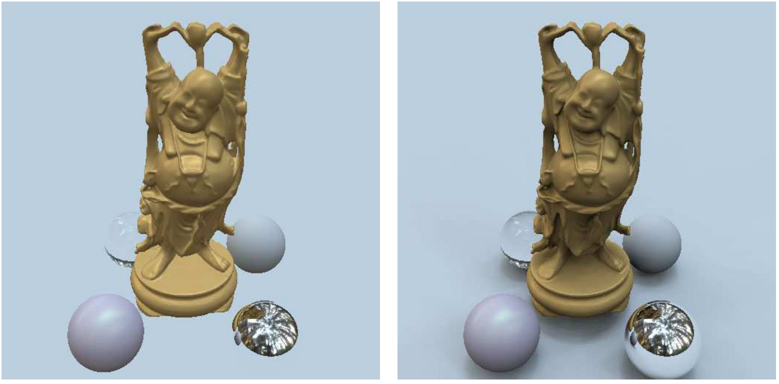
1. **No intersections**: the ray of a pixel does not hit an object in its traveral of the scene. In this case the colour of the pixel is set to the one of the environment map using the direction of the ray as look-up vector.
2. **Pure Specular**: the ray of a pixel intersects an object with a pure specular material. In this case the ray is reflected and/or refracted according to the material properties.
3. **General Material**: the ray of a pixel intersects an object with a general material described by a BRDF. In this case a modified Rendering Equation [91] is evaluated as:

   $$L(\mathbf{x},\omega) = L_e + \int_{\Omega} L(\omega') f_r(\omega',\omega) V(\mathbf{x},\omega') \mathbf{n} \cdot \omega' d\omega' \tag{2.80}$$

   where $\mathbf{x}$ and $\mathbf{n}$ are respectively the position and normal of the hit object, L_e is the emitted radiance at point $\mathbf{x}$, L is the environment map, f_r is the BRDF, ω' is the out-going direction, and ω is the view vector. V is the visibility function, a Boolean function that determines if a ray is obstructed by an object or not.

The use of ray tracing efficiently remove the inter-reflections/refractions limitations of the first approaches. Furthermore, visibility is evaluated allowing shadows and a more realistic visualisation, see Figure 2.45.

The evaluation of IBL using Equation 2.80 is computationally very expensive. For instance, if all directions stored in an environment map are used in the evaluation the complexity for a single pixel is $\mathcal{O}(nf(m))$ where n is the number of pixel of an environment map, and $f(m) =$



(a) (b)

Figure 2.45: *An example of IBL evaluating visibility applied to Stanford's Happy Buddha Model [73]: a) IBL evaluation without shadowing. b) IBL evaluation with Debevec's method [44].*

$\log m$ is the complexity for computing the visibility in a scene with m objects. To lower the complexity there are two methods: light source extraction or Monte-Carlo integration.

Light Source Extraction

A simple method for the evaluation of Equation 2.80 is to generate a finite set of directional light sources from the environment map. The idea is to place a light source in an area with high luminance values.

The main techniques are: structured importance sampling (SIS) [4], k-means sampling (KMS) [97], Penrose tiling sampling (PTS) [148], median-cut sampling (MCS) [170]. The main difference in all these methods is how lights are placed. In SIS, a light is placed in the centre of a stratum generated by k-centre on a segmented environment map. In KMS, lights are generated randomly on the environment map and then they are relaxed using Lloyds's method [119]. In PTS, the image is decomposed using Penrose tiles, where smaller tiles are applied to areas with high luminance, and a light source is placed for each vertex of the a tile. Finally in MCS, light sources are placed in the weighted centre of each region that the image is hierarchically decomposed by a 2D-tree, that divides regions on the total luminance power of it.

After the generation of light sources, Equation 2.80 is evaluated as:

$$L(\mathbf{x},\omega) = L_e + \sum_{i=1}^{N} C_i f_r(\omega'_i,\omega)(\mathbf{n}\cdot\omega'_i)V(\mathbf{x},\omega'_i) \tag{2.81}$$

where N is the number of generated light sources, $-\omega_i'$ is the direction of the generated light source, and C_i is the colour of it. For an example see Figure 2.46.

(a)

(b)

Figure 2.46: *An example of evaluation of Equation 2.81 using MCS [170] with different N: a) $N = 16$. Note that aliasing artifacts can be noticed. b) $N = 256$, aliasing is alleviated.*

The light sources extraction does not present noise, so they can be used in animated scenes. However, the method can present aliasing if only few light sources are generated, depending on radiance distribution and the dynamic range of the environment map.

Monte-Carlo Integration and Importance Sampling

Another popular method for IBL is to use Monte-Carlo integration, which uses random sampling for evaluating complex multidimensional integrals as in the case of Equation 2.80. For example, an 1D $f(x)$ function needs to be integrated in the domain $[a,b]$ is usually solved as:

$$I_{ab} = \int_a^b f(x)dx = F(a) - F(B) \quad F'(x) = f(x) \tag{2.82}$$

However, it can happen that $F(x)$ is not possible to integrate analytically as in the case of Normal distribution, or $f(x)$ is known only in few points of the domain. In Monte-Carlo integration [163], integration is calculated by averaging the value of $f(x)$ in N points distributed in a domain, assuming Riemann integrals:

$$\hat{I}_{ab} = \frac{b-a}{N}\sum_{i=1}^{N} f(x_i) \qquad I_{ab} = \lim_{N\to+\infty} \frac{b-a}{N}\sum_{i=1}^{N} f(x_i) \tag{2.83}$$

where $x_1, x_2, ..., x_N$ are random uniformly distributed points in $[a,b]$, because deterministic chosen points [163] do not work efficiently in the case of multidimensional integrals. Hence, to integrate a multidimensional function are needed equidistant point grids, which are very large N^d, where N is the number of points for a dimension and d is the number of dimensions of $f(x)$.

The convergence in the Monte-Carlo integration Equation 2.83 is determined by variance, $\sigma \propto N^{-\frac{1}{2}}$, which means that N has to be quadruplicated to half the error. A technique that reduces variance is called importance sampling. The idea is to integrate by taking points x_i that contribute more to the final integral. This is achieved by using a probability density function $p(x)$ with a similar shape of $f(x)$:

$$\hat{I}_{ab} = \frac{1}{N}\sum_{i=1}^{N}\frac{f(x_i)}{p(x_i)} \tag{2.84}$$


(a) (b)

Figure 2.47: *A comparison between Monte-Carlo integration methods for IBL: a) Monte-Carlo integration using 4 samples per pixel. b) Importance Sampling Monte-Carlo integration using 4 samples per pixel.*

Note that the variance is still the same, but a good choice of $p(x)$ can make it arbitrarily low. The optimal case is when $p(x) = \frac{f(x)}{I_{ab}}$. To create samples, x_i, according to $p(x)$ the inversion method can be applied. This method calculates the cumulative distribution function $P(x)$ of $p(x)$, then samples, x_i, are generated by $x_i = P^{-1}(y_i)$ where $y_i \in [0,1]$ is a uniformly random distributed number.

Importance sampling can be straightforwardly applied to the IBL problem, extending the problem to more than one dimension [162]. Good choices of $p(x)$ are the luminance of the envi-

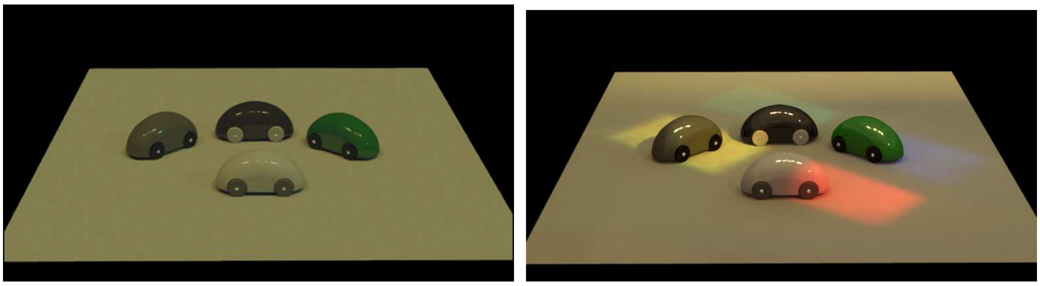
(a) (b)

Figure 2.48: *An example of stereo IBL by Corsini et al. [36] using the Michelangelo's David model: a) Three Michelangelo's rendered using a single environment map. b) Three Michelangelo's rendered using a stereo environment map. Note that shadows directions change for each model. Images are courtesy of Massimiliano Corsini.*

ronment map image, $l(\omega')$, or the BRDF, $f_r(\omega, \omega')$, or a combination of both. For an example of the evaluation of IBL using Monte-Carlo integration see Figure 2.47. Monte-Carlo methods are unbiased, they converge to the real value of the integration, but they have the disadvantage of noise, that can be alleviated with importance sampling.

2.7.3 Beyond Environment Map

The use of a single environment map allows only infinite distant illumination, and introduces parallax problems. Recently, some research has provided solutions to this problem.



(a) (b)

Figure 2.49: *An example of dense sampling methods by Jonas et al. [197, 196] a toy cars scene: a) The scene rendered using a single environment map. b) The scene rendered using lightfields. Note that more details of lighting are present. Images are courtesy of Jonas Unger.*

Corsini et al. [36] proposed to capturing two environment maps for each scene and to solve spherical stereo [114]. When the geometry of the scene is extracted, light sources are extracted and generated for use in the 3D scene, for an example see Figure 2.48.

A different approach was presented by Jonas et al. [197, 196] where the scene is densely sampled capturing environment maps with an HDR camera. Collected data is arranged in a lightfield [113, 70], and used for drawing samples using Monte-Carlo rendering techniques [162], for an example see Figure 2.49.

2.8 Summary

This chapter has presented a detailed overview of all the work conducted so far in the field of the HDR imaging: capturing, storing, visualisation, TMO evaluation, and IBL. The next chapter will review Inverse Tone Mapping, an emerging field in HDR imaging, which is the main topic of the thesis.

Chapter 3

Inverse Tone Mapping Operators

Inverse tone mapping is an emerging topic in the computer graphics community that links LDR imaging and HDR imaging. This is achieved by transforming LDR content, tone mapped or single exposure, into HDR content using an operator called inverse Tone Mapping Operator (iTMO). iTMOs can be classified in two kinds:

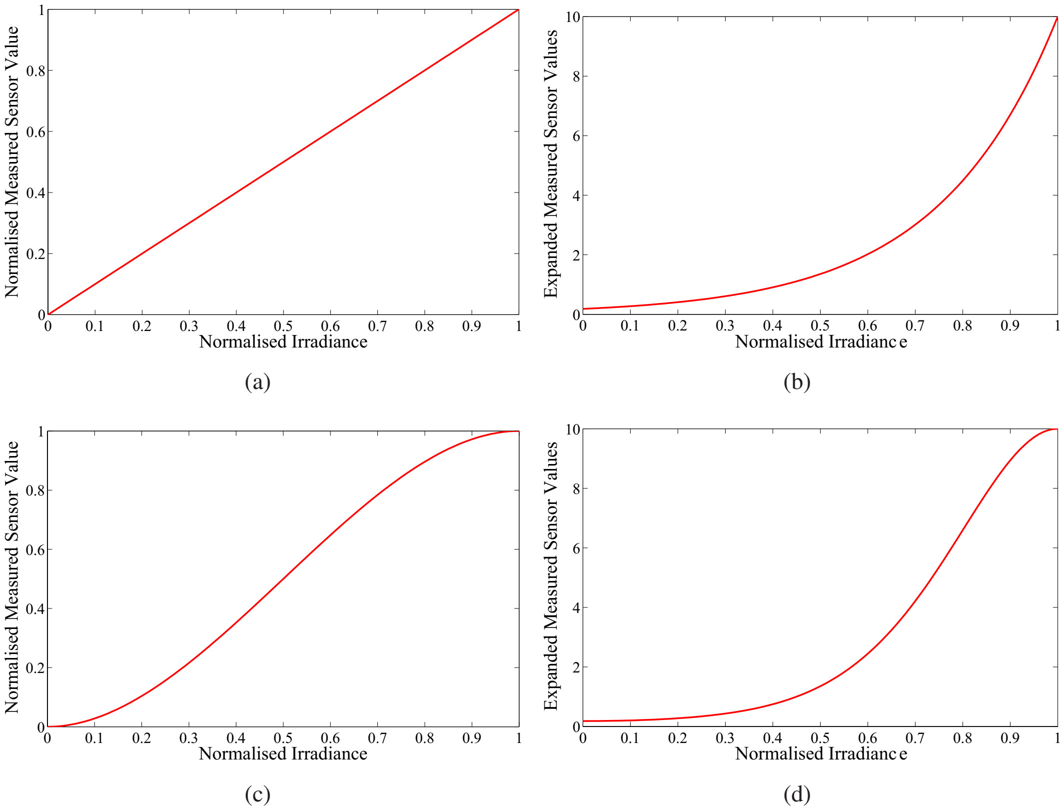
- **iTMOs for LDR Content Expansion**: expand LDR images/videos to create HDR content suitable for HDR monitors or IBL.
- **iTMOs for HDR Content Compression**: expand tone mapped content. This content was previously compressed using LDR standards.

3.1 Inverse Tone Mapping for Generating HDR Content from Single Exposure Content

The application of an iTMO to an LDR single exposure content is mainly the reconstruction of HDR content. This is an ill-posed problem, because information is missing in over-exposed and under-exposed regions of the image/frame.

3.1.1 Linearisation of the Signal using a Single Image

Linearisation of the signal is a very important step in many iTMOs in order to process further images. The main reason to work in a linear space is that there is more control and predictability over the expansion. In an unknown space, it is hard to predict how the expansion will behave, see Figure 3.1 for an example. Furthermore, precise/estimated measurements of scene radiance are needed to recover scene properties, such as mean, harmonic mean, standard deviation, and etc.



(a) (b)

(c) (d)

Figure 3.1: *An example of the need of working in the linear space: a) Measured values from a sensor in the linear domain. b) The expansion of signal in a) using* $f(x) = 10e^{4x-4}$*. c) Measured values from a sensor with an unknown CRF applied to it. d) The expansion of signal c) using* f*. Note that the expanded signal in d) has a different shape from b) which was the desired one.*

When there is access to the capturing device, linearisation can be done calculating the CRF of that device. This function can be calculated using methods presented in Section 2.2.2. Also in the ideal case when images are stored using raw data format (RAW) from the CCD sensor, this step is not needed, because RAW format stores linear values. However, in the general case, the CRF of the camera/video-camera is no available, and images/videos are stored in 8-bit non-linear format. In this case the estimation of the CRF needs to be carried out from a single image or a couple of frames.

Blind Inverse Gamma Function

Computer generated images or processed RAW photographs are usually stored with gamma correction. Farid [60] proposed an algorithm based on multi-spectral analysis to blindly linearise the signal when gamma correction is applied to the image.

Farid observed that the application of gamma to a signal introduces new harmonics. For example given a simple linear signal:

$$y(x) = a_1 \sin(\omega_1 x) + a_2 \sin(\omega_2 x) \tag{3.1}$$

if $y(x)$ is gamma corrected and it is rewritten using a second term Taylor expansion, $y(x)_2^\gamma$, the result is given by:

$$\begin{gathered} y(x)^\gamma \approx y(x)_2^\gamma = a_1 \sin(\omega_1 x) + a_2 \sin(\omega_2 x) + \\ +\frac{1}{2}a_1^2(1+\sin(2\omega_1 x)) + \frac{1}{2}a_2^2(1+\sin(2\omega_2 x)) + 2a_1 a_2(\sin(\omega_1+\omega_2)+\sin(\omega_1-\omega_2)) \end{gathered} \tag{3.2}$$

As can be seen, new harmonics are introduced after gamma correction which are correlated with the original ones: $2\omega_1$, $2\omega_2$, $\omega_1+\omega_2$, and $\omega_1-\omega_2$.

High order correlations introduced by the gamma in a one-dimensional signal $y(x)$ can be estimated with high order spectra, for example a normalised bi-spectrum correlation estimates third order correlations [130]:

$$b^2(\omega_1, \omega_2) = \frac{E[Y(\omega_1)Y(\omega_2)\overline{Y}(\omega_1+\omega_2)]}{E[|Y(\omega_1)Y(\omega_2)|^2]E[|Y(\omega_1+\omega_2)|^2]} \tag{3.3}$$

where Y is the Fourier transform of y. $b^2(\omega_1, \omega_2)$ can be estimated using overlapping segments of the original signal y:

$$\hat{b}(\omega_1,\omega_2) = \frac{|\frac{1}{N}\sum_{k=0}^{N-1} Y_k(\omega_1)Y_k(\omega_2)\overline{Y}_k(\omega_1+\omega_2)|}{\sqrt{\frac{1}{N}\sum_{k=0}^{N-1}|Y_k(\omega_1)Y_k(\omega_2)|^2\frac{1}{N}\sum_{k=0}^{N-1}|Y_k(\omega_1+\omega_2)|^2}} \tag{3.4}$$

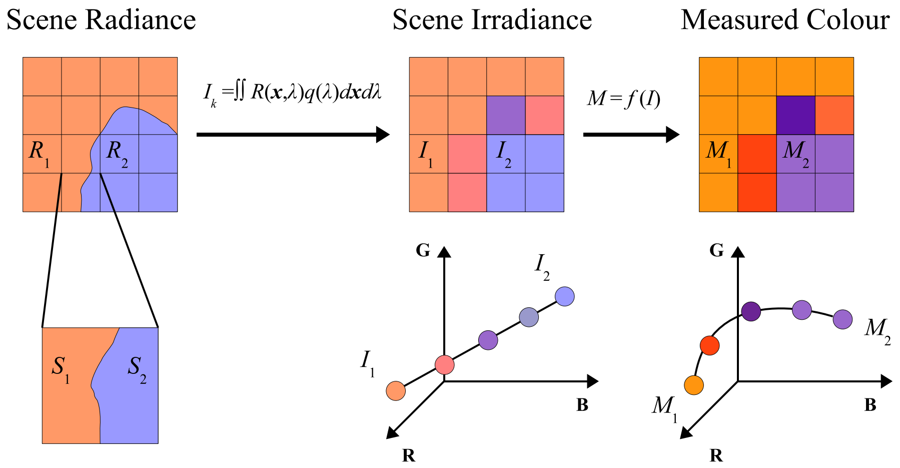
where Y_k is the Fourier transform of the k-th segment, and N is the number of segments. Finally, the gamma of an image is estimated by applying a range of inverse gamma to the gamma corrected image and choosing the value that minimise the measure of the third order correlations as:

$$\varepsilon = \sum_{\omega_1=-\pi}^{\pi}\sum_{\omega_2=-\pi}^{\pi}|\hat{b}(\omega_1,\omega_2)| \tag{3.5}$$

Farid evaluated this technique with fractal and natural images, and showed that the recovered γ values had respectively an average errors of 5.3% and 7.5%.

Radometric Calibration from a Single Image

A general approach for linearisation using a single image was proposed by Lin et al. [116]. This approach exploits information at edges of the image for approximating the CRF.



(a) (b) (c)

Figure 3.2: *A coloured edge region: a) The real scene radiance value and shape at the edge with two radiance values R_1 and R_2. b) The irradiance pixels which are recorded by the camera/video-camera with interpolated values between I_1 and I_2. Note that colours are linearly mapped in the RGB colour space. c) The measured pixels after the CRF application, in this case colours are mapped on a curve in the RGB colour space.*

In an edge region Ω there are mainly two colours I_1 and I_2 separated by the edge. Colours between I_1 and I_2 are a linear interpolation of these two, and form a line in the RGB colour space, see Figure 3.2.b. When the CRF is applied to these values, $M = f(I)$, since it is a general non-linear function, the line of colours is transformed into a curve, see Figure 3.2.c. In order to linearise the signal the inverse CRF, $g = f^{-1}$, has to map measured values $M(\mathbf{x})$ to a line defined by $g(M_1)$ and $g(M_2)$. This is solved by finding a g that for each pixel $M(\mathbf{x})$ in the region, Ω, minimises distance from $g(M(\mathbf{x}))$ to the line $\overline{g(M_1)g(M_2)}$:

$$D(g,\Omega) = \sum_{\mathbf{x}\in\Omega} \frac{|(g(M_1) - g(M_2)) \times (g(M(\mathbf{x})) - g(M_2))|}{|g(M_1) - g(M_2)|} \tag{3.6}$$

Edge regions, which are suitable for the algorithm, are chosen from non overlapping 15×15 windows with two different uniform colours along edges, which are detected using a Canny filter [69]. The uniformity of two colours in a region is calculated using variance. To improve the quality and complete the missing parts of g, a Bayesian learning step is added using inverse CRFs from real cameras [72].

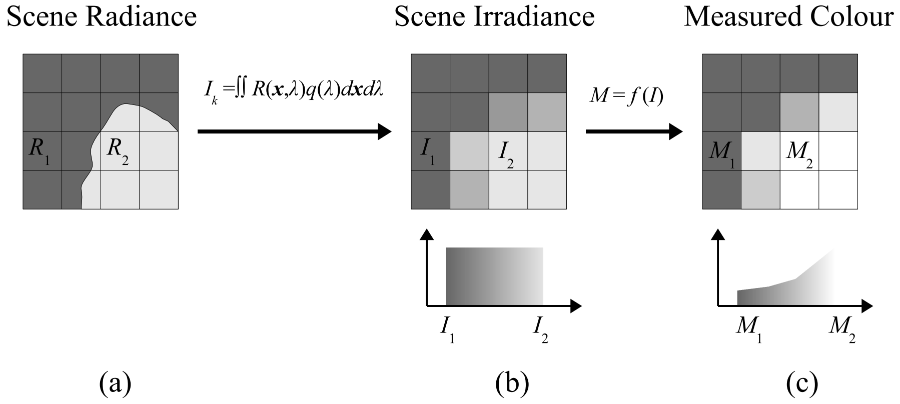


Figure 3.3: *A grey-scale edge region: a) The real scene radiance values and shape at the edge with two radiance values R_1 and R_2. b) The irradiance pixels which are recorded by the camera/video-camera. Irradiance values are uniformly distributed, this results in an uniform histogram. c) The measured pixels after the CRF application. This transforms the histogram from an uniform one into a non-uniform histogram.*

Lin and Zhang [117] extended Lin et al.'s method [116] to grey-scale images. In this case g is estimated using a function that maps non-uniform histograms of M into uniform ones, see Figure 3.3. They propose a measure for determining the uniformity of an histogram H of an image, which is defined as:

$$N(H)=\frac{1}{b}\sum_{k=I_{\min}}^{I_{\max}}\left[|H(k)|-\frac{|H(k)|}{b}\right]^2+\frac{\beta}{3}\sum_{n=1}^{3}\left[|H_n|-\frac{|H|}{3}\right]^2 \quad \text{where } H_n=\sum_{i=I_{\min}+\frac{(n-1)b}{3}}^{I_{\min}+\frac{nb}{3}-1}|H(i)| \tag{3.7}$$

where $|H|$ is the number of pixels in H, $|H(k)|$ is the number of pixels of intensity k, b is the number of grey scale levels, β is an empirical parameter, and $I_{\min}$ and $I_{\max}$ are respectively the minimum and maximum intensity in H. The inverse CRF is calculated similarly as in Lin et al.'s method [116], where g is a function that minimises the histogram uniformity for each edge region:

$$D(g,\Omega)=\sum_{H\in\Omega}\omega_H N(g(H)) \tag{3.8}$$

where $\omega_H=\frac{|H|}{b}$ is a weight for giving more importance to dense histogram. Regions are chosen as in Lin et al. [116], and g is refined with a Bayesian learning. The method can be applied to coloured images, applying it to each colour channel.

3.1.2 Bit Depth Expansion for High Contrast Monitors

Daly and Feng [41, 42] proposed a couple of methods for extending the bit-depth of 8-bit images and videos (6-bit for MPEG-2 videos). New LCD monitors present high contrast, on average around 1,000:1, and high luminance peak, on average around 400 cd/m^2. This means that displaying 8-bit data, without any refinement, would entail having the content linear expanded for higher contrast visualised as artifacts such as banding/contouring. The goal of their methods is to create a medium dynamic range image, removing contouring in the transition areas, without a particular interest for over-exposed and under-exposed areas.

Amplitude Dithering for High Contrast Displays

The first algorithm [41] is based on amplitude dithering by Roberts [173], see Figure 3.4. Amplitude dithering or noise modulation is a straightforward dithering technique which adds a noise pattern to an image before quantisation. When the image needs to be visualised, the pattern is removed from it. The bit depth is perceived higher than the real one, because there is a subsequent averaging happening in the display and in the HVS.

Daly and Feng modified Roberts' technique to apply it to high contrast displays. Subtractive noise was employed instead of additive, since during visualisation a monitor can not remove it. The authors modelled the noise combining the effect of fixed pattern display noise and the one perceived by the HVS, making the noise invisible. They used the contrast sensitivity function (CSF) which is a 2D and anisotropic function derived by psychophysical experiments [40]. The CSF can be extended in the temporal dimension [217] for moving images, which allows the noise to have a higher variance, and furthermore, they show that the range can extend by an extra bit.

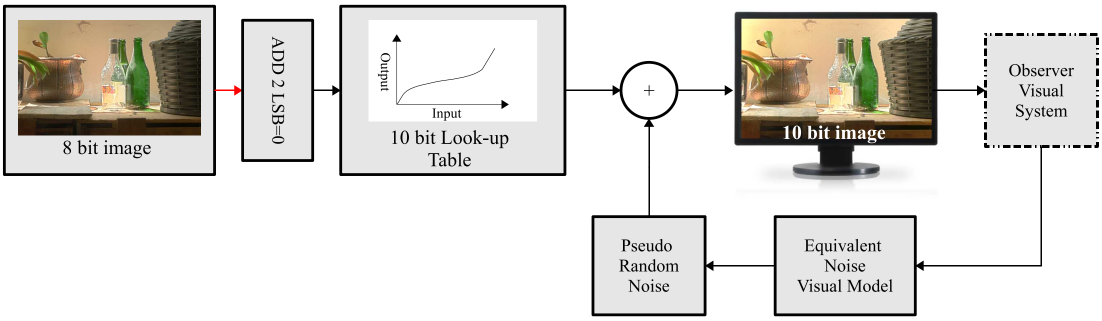


Figure 3.4: *The pipeline for bit depth extension using amplitude dithering in Daly and Feng's method [41].*

Contouring Removal

The second algorithm [42] presents a different approach where contours are removed instead of being masked with invisible noise. The first step of the algorithm is to filter the starting image at p-bit using a low-pass filter, see Figure 3.5. The filter needs to be wide enough to span across the false contours. Note that this operation increases the bit depth to $n > p$ because during averaging a higher precision is needed than the one for the original values. Then this image is quantised at p-bit, where any contour that appears here are false one, because the image has no high frequencies. Subsequently, the false contours are subtracted from the original image, and the filtered image at p-bit is added for restoring low frequency components. The main limitation of the algorithm is that it keeps artefacts in high frequencies, but they are hard to detect by HVS due to frequency masking.

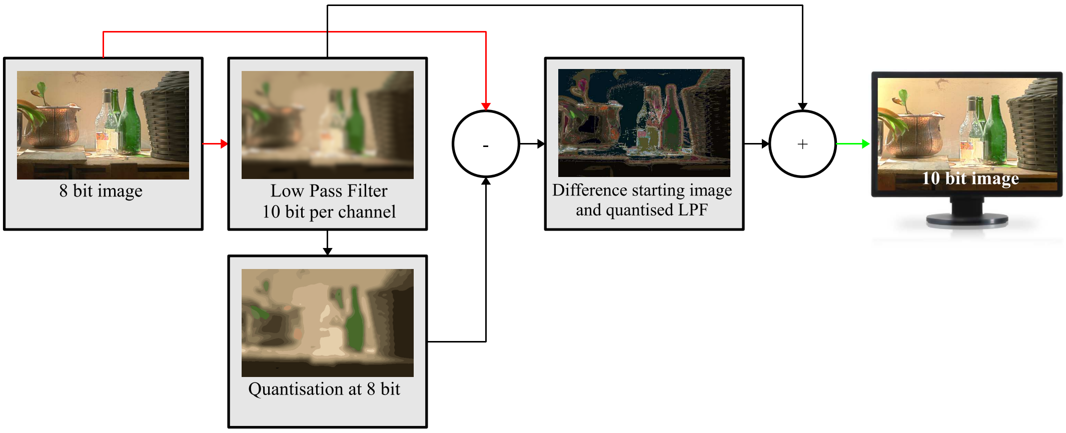


Figure 3.5: *The pipeline for bit depth extension using de-contouring in Daly and Feng's method [42].*

3.1.3 A Power Function Model for Range Expansion

One of the first iTMOs was proposed by Landis [105] for IBL in the visual effects industry. The operator is based on power functions, and expands the luminance of a LDR image as:

$$L_{\mathrm{w}}(\mathbf{x}) = \begin{cases} (1-k)L_{\mathrm{d}}(\mathbf{x}) + kL_{\mathrm{w,\,Max}}L_{\mathrm{d}}(\mathbf{x}) & \text{if } L_{\mathrm{d}}(\mathbf{x}) \geq R \\ L_{\mathrm{d}}(\mathbf{x}) & \text{otherwise} \end{cases} \qquad k = \left(\frac{L_{\mathrm{d}}(\mathbf{x})-R}{1-R}\right)^{\alpha} \tag{3.9}$$

where R is the threshold for expansion ($R = 0.5$ in the original work), $L_{\mathrm{w,\,Max}}$ is the maximum luminance of the expanded image, and α is the exponent of fall-off that controls the stretching curve.

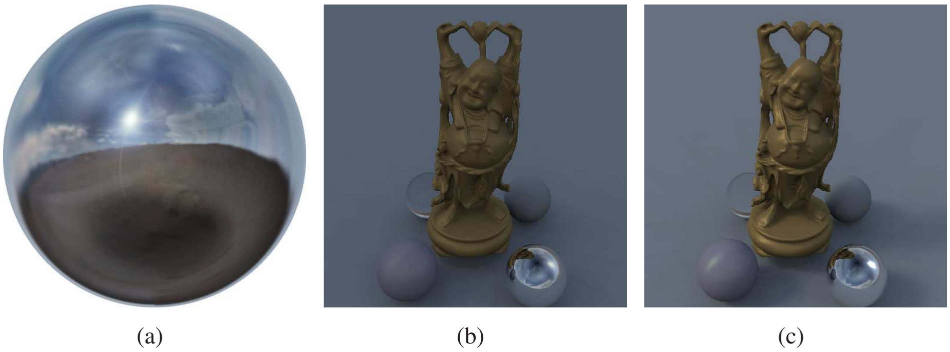


Figure 3.6: *An example of re-lighting of the Stanford's Happy Buddha using Landis' method [105]: a) An LDR environment map. b) The re-lighted scene using the LDR environment in a). c) The re-lighted scene using the expanded version of the LDR environment map in a).*

While this technique produces suitable HDR environment map for IBL, it may not produce good quality images/videos that can be visualised on HDR monitors. This is because it does not handle artefacts such as exaggeration of compression or quantisation artefacts.

3.1.4 Highlight Generation for HDR Monitors

Meylan et al. [132, 133] presented an iTMO with the specific task of reproducing highlights in LDR images when displayed on HDR monitors. The main concept is to detect the diffuse and specular part of the image and to expand these using different linear functions. The detection is based on the assumption that highlights are small and bright, which means that the maximum diffuse luminance value ω is obtained as the maximum of the low-pass filtered luminance channel L_d of the image. However, more processing is needed to avoid the case when white diffuse regions are next to highlights regions, see Figure 3.7 for the complete pipeline for calculating ω.

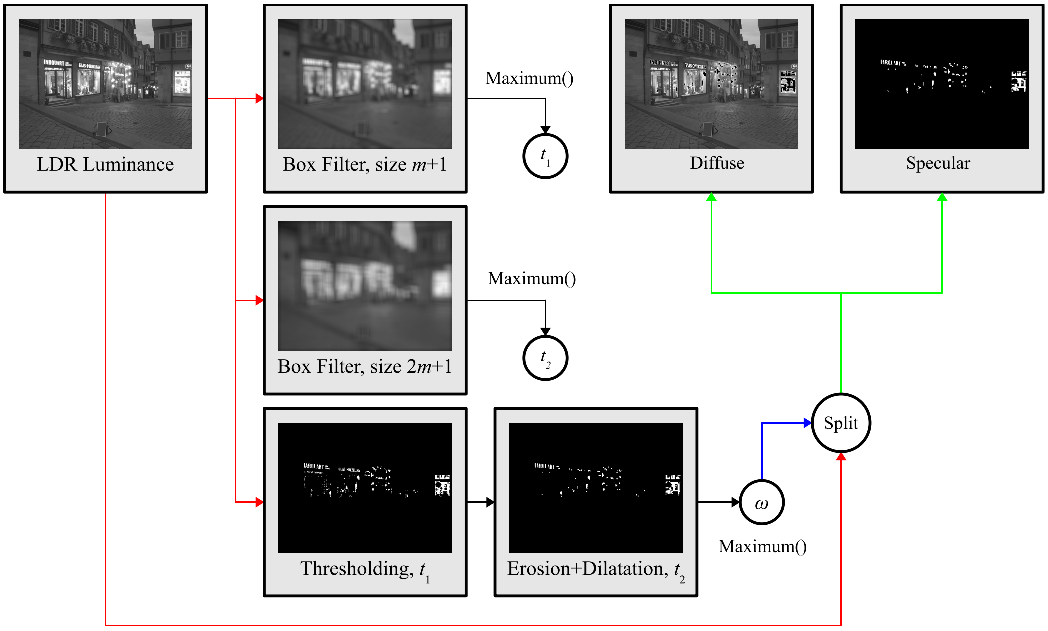


Figure 3.7: *The pipeline for calculating the maximum diffuse luminance value ω in an image in Meylan et al.'s method [133]. The image is firstly filtered using a box filter of size m to calculate value t_1 as maximum the luminance value. This operation is repeated for a filter of size $2m+1$ to calculate t_2. Then, t_1 is used as threshold on the original luminance resulting in a mask, subsequently an erosion and dilation filter is applied to the mask using t_2. While pixels active in the mask are considered specular pixels, black pixels are considered as diffuse ones.*

After the calculation of ω, the luminance channel is expanded using the following function:

$$f(L_{\mathrm{w}}(\mathbf{x})) = \begin{cases} s_1 L_{\mathrm{d}}(\mathbf{x}) & \text{if } L_{\mathrm{d}}(\mathbf{x}) \leq \omega \\ s_1\omega + s_2(L_{\mathrm{d}}(\mathbf{x}) - \omega) & \text{otherwise} \end{cases} \qquad s_1 = \frac{\rho}{\omega} \quad s_2 = \frac{1-\rho}{L_{\mathrm{d,\,Max}} - \omega} \tag{3.10}$$

where $L_{\mathrm{d,\,Max}} = 1$ since the image is normalised, and ρ is the percentage of the HDR display luminance allocated to the diffuse part. However, a global application of f can produce quantisation artefacts around the enhanced highlights. This is reduced using a low pass filter only in the expanded regions, see Figure 3.8.

Finally, series of psychophysical experiments were run to determine the value of ρ for f using the DR-37P Dolby HDR monitor [49]. The results showed that for outdoor scenes users preferred a high value of ρ, which means a small percentage of dynamic range allocated to highlights, while for indoor scenes this is the contrary. For indoor and outdoor scenes of equal diffuse brightness users chose a low value for ρ, so they preferred more range allocated to highlights. In conclusion from the analysis of the data, $\rho = 0.66$ is a good estimate in general.

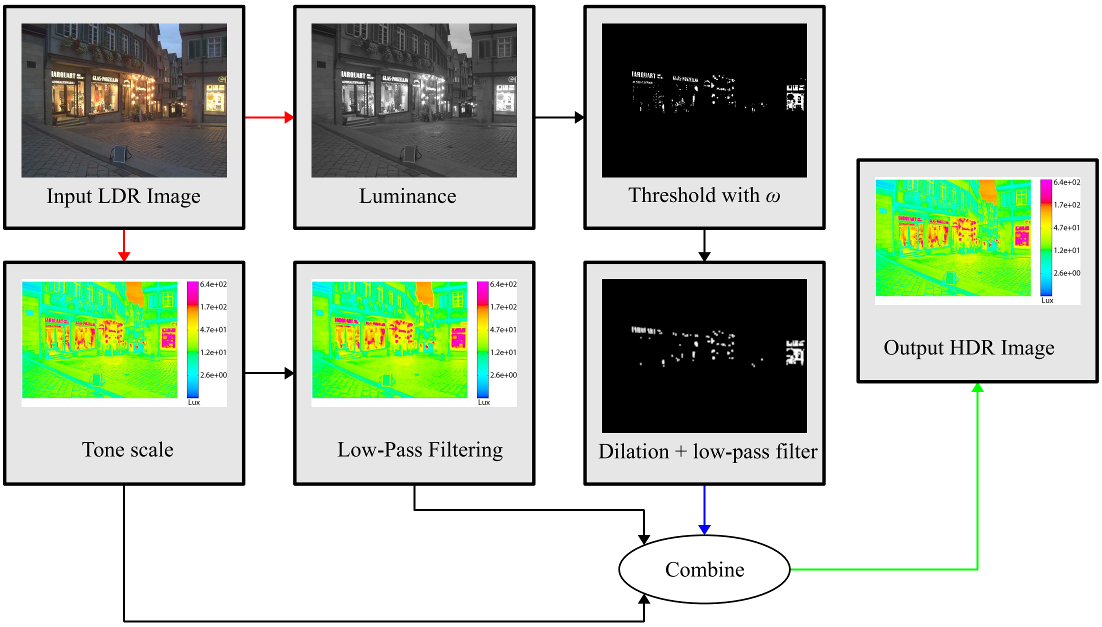


Figure 3.8: *The pipeline for the range expansion in Meylan et al. [133]. The original LDR image is expanded using Equation 3.10. Then, expanded luminance is filtered using a low pass filter. Finally, filtered expanded luminance and unfiltered one are linearly interpolated using a mask. This mask is calculated by thresholding LDR luminance with ω. To remove noise, the mask is filtered with a dilatation and low pass filter.*

This algorithm is designed for the reproduction of highlights on HDR monitors. The use for other tasks, such as enhancement of videos, needs more processing and a classifier, which was underlined by authors' evaluation experiment. Furthermore, there is no linearisation stage which can result in an exaggerated gamma effect.

3.1.5 Hallucination of HDR Images

A different approach to inverse tone mapping problem was proposed by Wang et al. [207], where they showed the benefits of an in-painting system to recover lost details in over-exposed and under-exposed regions of an image, combined with a luminance boosting. The whole process is called *hallucination*, and their system presents a mixture between automatic and user-based approaches.

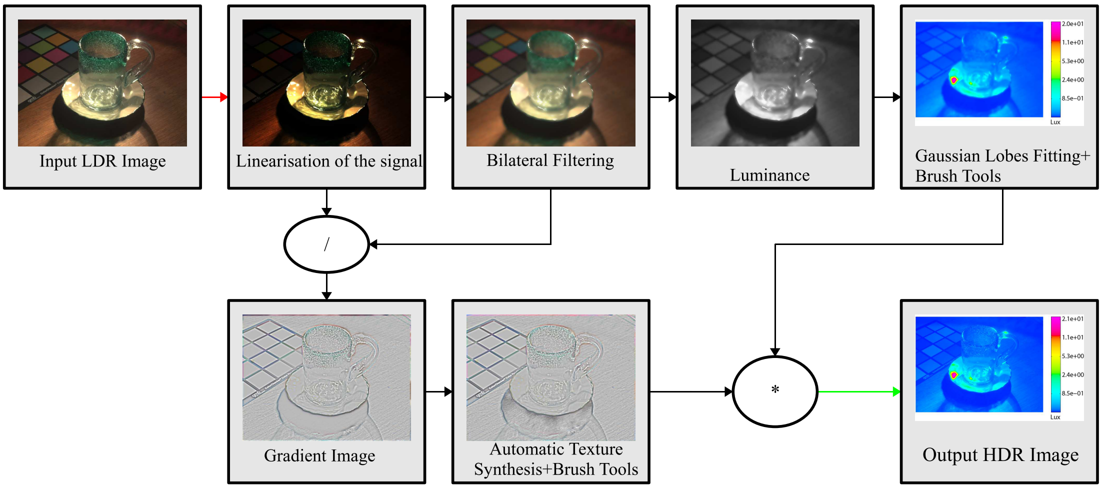


Figure 3.9: *The pipeline of Hallucination of HDR in Wang et al.'s method [207].*

The first step of hallucination is to linearise the signal, see Figure 3.9 for the complete pipeline. This is achieved with an inverse gamma function with $\gamma = 2.2$, which is the standard value for DVDs and television formats [89]. After this step, the image, I, is decomposed into large the scale illumination layer, I_f, and the fine texture detail layer, I_d, using the bilateral filtering.

Radiance in I_f is reconstructed using a linear interpolation of elliptical Gaussian kernels. Firstly, a weight map, w, is calculated for each pixel as:

$$w(\mathbf{x}) = \begin{cases} \frac{C_{ue} - Y(\mathbf{x})}{C_{ue}} & Y(\mathbf{x}) \in [0, C_{ue}) \\ 0 & Y(\mathbf{x}) \in [C_{ue}, C_{oe}) \\ \frac{Y(\mathbf{x}) - C_{oe}}{1 - C_{oe}} & Y(\mathbf{x}) \in [C_{oe}, 1] \end{cases} \tag{3.11}$$

where $Y(\mathbf{x}) = R_{\mathrm{d}}(\mathbf{x}) + 2G_{\mathrm{d}}(\mathbf{x}) + B_{\mathrm{d}}(\mathbf{x})$, and C_{ue} and C_{oe} are respectively the threshold for under-exposed and over-exposed pixels. The authors suggested values of 0.05 and 0.85 for C_{ue} and C_{oe} respectively. Secondly, each over-exposed region is segmented and fitted with an elliptical Gaussian lobe G, where variance of the axis is estimated using region extents. The profile of G is calculated using an optimisation procedure based on non over-exposed pixels at the edge of the region. Finally, the luminance is blended using a simple linear interpolation:

$$O(\mathbf{x}) = w(\mathbf{x})G(\mathbf{x}) + (1 - w(\mathbf{x}))\log_{10} Y(\mathbf{x}) \tag{3.12}$$

Users can optionally add Gaussian lobes using a brush. The lost texture details in I_d are reconstructed using a texture synthesis technique similar to [20]. However, this automatic synthesis has some limits when scene understanding is needed, therefore a warping tool was included. This allows the user to select with a stroke-based interface a source region and a target region, and texture details will be transferred. A similar tool is available in Adobe Photoshop [3], and it is called stamp tool.

Finally, the two reconstructed layers, I_d and I_f, are blended together using Poisson blending [159] obtaining the final hallucinated or expanded image.

This system can be used for both IBL and visualisation of images, allowing the reconstruction of lost details in clamped regions. However, the system can process only single images and not videos, because user's intervention is needed. Moreover, there is the possibility to not find information to transfer in the same image or in other images.

3.1.6 LDR2HDR

A straightforward and fast technique was proposed by Rempel et al. [171]. Their goal was a real-time iTMO for DVDs and television program videos.

The algorithm pipeline is shown in Figure 3.10. The first step of the LDR2HDR algorithm is to remove artefacts due to compression algorithms such as MPEG of the media using bilateral filter. A sophisticated artefact removal algorithm is not employed due to real-time constraints. Then, the signal is linearised employing a simple inverse gamma, because the method

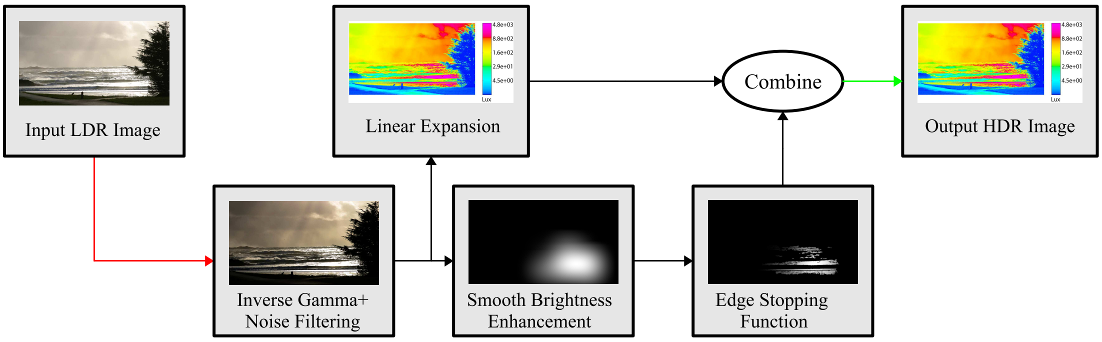


Figure 3.10: *The pipeline of Rempel et al.'s method [171]. Firstly, gamma is inverted and filtering for the linearisation of signal and reduction of noise. Secondly, the range is linearly expanded. In parallel, a BEF is created to increase contrast in high luminance areas, where edges are enhanced using an edge-stop function. Finally, the linearly expanded image is enhanced using the BEF.*

is meant for DVDs and television programs as Hallucination [207]. Afterwards, the contrast is stretched in an optimised way for the Dolby DR-37P HDR monitor [49]. A simple linear contrast stretching is applied to boost values, however the maximum contrast is limited to 5,000:1 to avoid artefacts. While the minimum value was mapped to 0.015 cd/m^2, the maximum one was mapped to 1,200 cd/m^2.

To enhance brightness in bright regions a Brightness Enhance Function (BEF) is employed. This function is calculated applying a threshold of 0.92 (on a scale $[0,1]$ for LDR values). At this point the image is Gaussian filtered using a filter with $\sigma = 30$ (150 pixels) which is chosen for $1,920 \times 1,080$ content. In order to increase contrast around edges, an edge stopping function is used. Starting from saturated pixels, a flood-fill algorithm strategy is applied until an edge is reached, which is estimated using gradients. Subsequently, a morphological operator is applied to remove noise, which is followed by a Gaussian filter with $\sigma = 1$. Finally, the BEF is mapped in the interval $[1, \alpha]$ where $\alpha = 4$, and it is multiplied with the scaled image to generate the final expanded image, see Figure 3.11 for an example. To improve efficiency the BEF is calculated using Laplacian pyramids [24] on the GPU or FPGA [49].

The algorithm was evaluated using HDR-VDP [127, 124] for comparing the linearised starting image with the generated HDR image. This evaluation showed that the proposed method does not introduce spatial artefacts during expansion of the content. Note that LDR2HDR processes each frame separately which may be not temporally coherent due to the nature of the BEF. More important is that the maximum peak of the DR-37P HDR monitor is 3,000 cd/m^2 meaning that

(a) Original LDR image (b) f-stop -6 (c) f-stop -8

(d) f-stop -10 (e) f-stop -12

Figure 3.11: *Application of Rempel et al.'s method [171] to the Beach image from the authors' website: a) The original LDR image. b,c,d,e) The different f-stops after expansion of a). The original LDR image is courtesy of Wolfgang Heidrich [171].*

LDR2HDR can produce HDR frames not suitable for the HDR monitor. $\alpha = 4$ and having boosted pixels with a maximum peak of 1,200 cd/m^2 can lead to a maximum peak of 4,800 cd/m^2 which has to be clamped by DR-37P.

3.1.7 Linear Scaling for HDR Monitors

A series of psychophysical experiments were run by Akyüz et al. [7] to evaluate tone mapped images, single exposure images and HDR images using the Dolby DR-37P HDR monitor [49]. The experiment involved 22 naïve participants between 20 and 40 years old, and in all experiments 10 HDR images were used, ranging from outdoor to indoor, from dim to very bright light conditions. The HDR images had around 5 orders of magnitude in order to be mapped to the HDR monitor.

The first experiment was a comparison between HDR and LDR images produced using various TMOs [107, 55, 169], an automatic exposure (that minimises the number of over/under-exposed pixels), and an exposure chosen by subjects in a pilot study. Images were displayed on the DR-37P using calibrated HDR images. LDR images were calibrated to match the appearance on a Dell UltraSharp 2007FP 20.1" LCD monitor on the HDR monitor. Subjects had the task of ranking images which were looking best to them. For each original test image a subject had to watch a trial image for 2 seconds which was randomly picked between the different type of images. The results of experiments showed that participants preferred HDR images, and between tone mapped images and single exposure (automatic and chosen by the pilot) there is not a large difference.

In the second experiment the authors compared expanded single exposure images with HDR images and single exposure images (automatic and chosen by the pilot). They employed a simple expansion method, which is defined as:

$$L_{\mathrm{w}}(\mathbf{x}) = k\left(\frac{L_{\mathrm{d}}(\mathbf{x}) - L_{\mathrm{d,\,Min}}}{L_{\mathrm{d,\,Max}} - L_{\mathrm{d,\,Min}}}\right)^{\gamma} \tag{3.13}$$

where k is the maximum luminance intensity of the HDR display, and γ is the non-linear scaling factor. For this experiment images with different γ values equal to 1, 2.2 and 0.45 were generated. The setup and the ranking task was the same as the first experiment. The results showed that brighter chosen exposure expanded images were preferred to HDR images, and viceversa when they had the same mean luminance. Authors suggested that mean luminance is more preferable than contrast. Finally, another important result is that linear scaling, $\gamma = 1$, was the most favoured expansion, suggesting that a linear scaling is maybe enough for an HDR experience.

The authors worked only with high resolution HDR images, without compression artefacts, and artistically captured. While this works well under such ideal conditions, in more realistic scenario such television programs or DVDs, where compression is employed, this makes not always the case. In these cases a more accurate expansion needs to be done in order to avoid amplification of compression artefacts, and contouring.

3.1.8 Enhancement of Bright Video Features for HDR Display

Didyk et al. [48] proposed an interactive system for enhancing brightness of LDR videos, targeting and showing results for DVD content. The main idea of the system is to classify a scene in three components: diffuse, reflections, and lights, and then to enhance only reflections and lights. The authors explained that diffuse components are difficult to enhance without creating visual artefacts and it was probably the intention of film-maker to show them saturated contrary to light sources and clipped reflections. The system works on non-linear values, because the goal is the restoration and not physical accuracy.

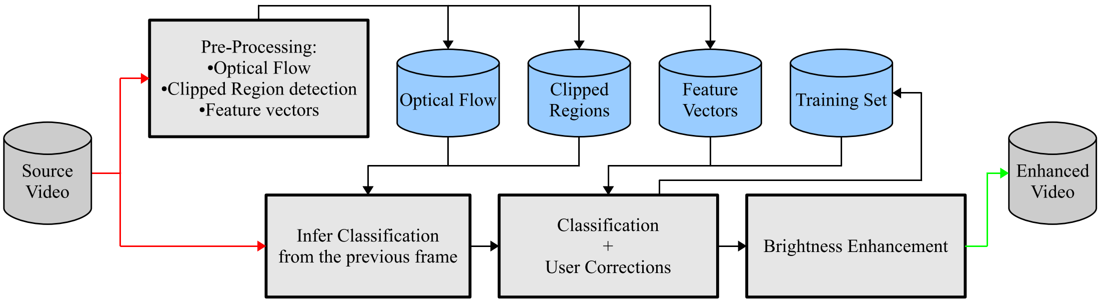


Figure 3.12: *The pipeline of the system proposed by Didyk et al. [48]: pre-processing (calculation of features vector, optical flow, and clipped regions), classification of regions using temporal coherence and a training set, user corrections (with updating of the training set), and brightness enhancement.*

The system consists of three main parts: pre-processing, classification, enhancement of clipped regions, see Figure 3.12 for the pipeline. The pre-processing step generates data needed during the classification. In particular it determines clipped regions using a flood-fill algorithm for luma values over 222 (230 for DVD content). Also, optical flow is calculated as well as other features such as image statistics, geometric features and neighbourhood characteristics.

Classification determines lights, reflections, and diffuse regions in a frame and relies on a training set of 2,000 manually classified regions. In the first part, a support vector machine [200] with kernel $k(z,z') = \exp(-\gamma\|z-z'\|^2)$ does an initial classification of regions. In the second part, motion tracking improves the initial estimation, using a nearest neighbour classifier based on a simple Euclidean metric:

$$d^2((z,x,t),(z',x',t')) = 50\|x-x'\|^2 + \|z-z'\|^2 + 5(t-t')^2 \tag{3.14}$$

where z are region features, x are coordinates in the image, and t is the frame number. The whole classification method had on average a classification error of 12.6% during tests. Tracking of clipped regions using motion compensation further reduced the percentage of objects that require manual correction to 3%. Finally, the user can supervise classified regions, correcting wrong classifications using an intuitive user interface, see Figure 3.13.

Clipped regions are enhanced by applying a non-linear adaptive tone curve, which is calculated based on partial derivatives within a clipped region stored in a histogram H. The tone curve is defined as an histogram equalisation on the inverted values of H:

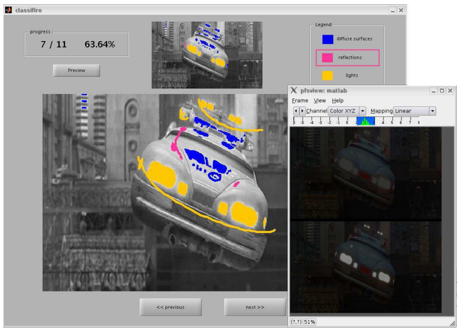


Figure 3.13: *The interface used for adjusting classification results of the system proposed by Didyk et al. [48]. The image is courtesy of Piotr Didyk [48].*

$$f(b) = k \sum_{j=2}^{b} (1 - H[j]) + t_2 \tag{3.15}$$

where t_2 is the lowest luma value for a clipped region, k is a scale factor that limits to the maximum boosting value m (equal to 150% for lights and 125% for reflections) and it is defined as:

$$k = \frac{m - t_2}{\sum_{j=1}^{N} (1 - H[j])} \tag{3.16}$$

where N is the number of bins in H. To avoid contouring during boosting, the luma channel is filtered with bilateral filtering separating it into fine details and a base layer, which will be merged after luma expansion.

Enhanced videos generated with this method were compared with original videos and ones processed using Rempel et al. [171], the only method at the time suitable for videos. Comparisons were performed running a paired comparison psychophysical experiment with 14 naïve subjects using a LCD Barco Coronis Color 3MP Diagnostic Luminance. The participants ranked 9 videos times 3 combinations: original video, Rempel et al. and their method. The study was analysed with a similar approach as in Ledda et al. [146]. The experiment showed that on overall scores this method was preferred with statistical significance compared to both original video and the Remepel et al. one. However, there is no statistically significance between this method and other ones for six of the analysed videos.

3.2 HDR Compression using Tone Mapping and Inverse Tone Mapping

Inverse tone mapping can be employed not only for the generation of content from a single exposure image, but also for HDR content compression. Tone mapping already compresses HDR content for visualising it on LDR monitors. An additional compression of HDR images can be achieved using traditional compression method such as JPEG, in the case of images or MPEG in the case of videos. In this, inverse tone mapping plays the role of the decoder to restore the HDR signal from an LDR one.

This approach to the compression of HDR content has the advantage of re-using previous compression schemes and standards. Also, it can allow backward-compatibility because the iTMO function can be easily stored in an extension header of a standard. These functions require a few parameters to be stored.

3.2.1 Backward Compatible JPEG-HDR

JPEG-HDR is an extension of JPEG compression scheme to HDR images by Ward and Simmons [213, 214]. The method does not use an explicit inverse tone mapping operator, nevertheless a spatial inverse function called Ratio Image (RI) is employed.

The encoding, see Figure 3.14, starts with the tone mapping of the HDR image. After this, the original HDR image is divided by the tone mapped one obtaining the RI which is going to be stored as a sub-band. The RI can be down-sampled reducing the sub-band size, because the HVS has a limited ability to detect large and high frequency changes in luminance. This fact was also exploited in Setzeen et al. [179] to improve the efficiency of HDR displays. However, down-sampling needs correction of the image, because the naïve multiplication of a down-sampled image times the tone mapped LDR image can produce halos/glare around edges. This problem can be solved in two ways: pre-correction and post-correction. The former method introduces corrections in the tone mapped image. This is achieved by down-sampling and afterwards up-sampling the RI image obtaining RI_d. Subsequently, the original HDR image is divided by RI_d, which is a tone mapped image with corrections. The latter method consists of

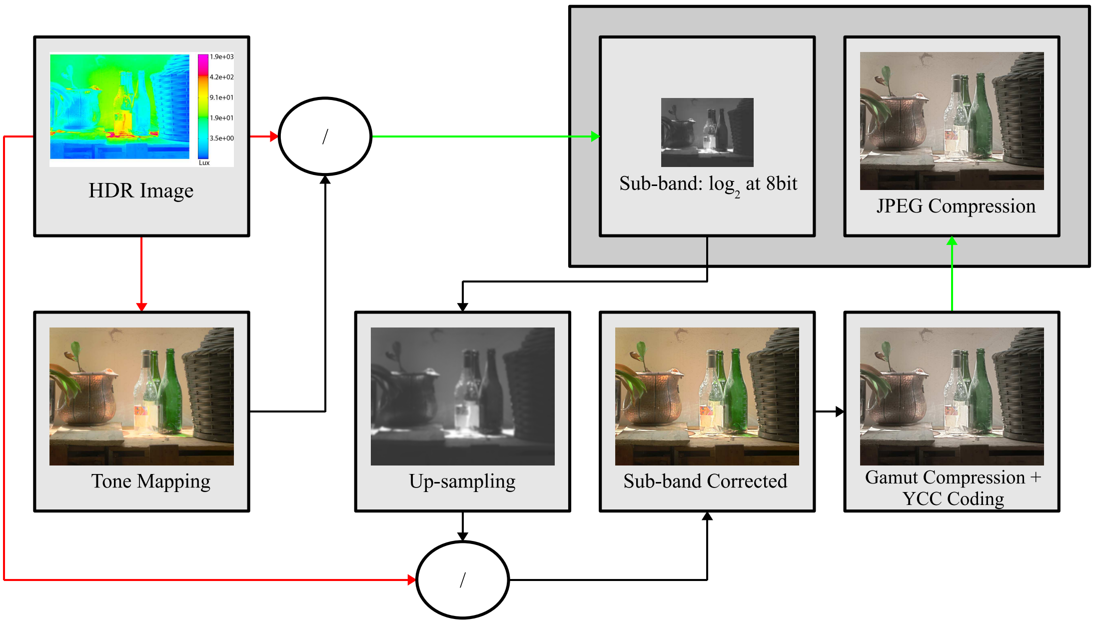


Figure 3.14: *The encoding pipeline for JPEG-HDR by Ward and Simmons [213, 214].*

an up-sampling with guidance, such as joint bilateral up-sampling [98], but it is more expensive than the pre-correction one. While RI_d is discritised at 8-bit in the logarithmic and stored in application markers of JPEG, the tone mapped layer needs a further processing for preserving colours. Two techniques are employed to solve this problems: compression of the gamut and a new $\mathrm{YC}_b\mathrm{C}_r$ encoding. A global desaturation is performed for the gamut compression. Given the following definition of saturation:

$$S(\mathbf{x}) = 1 - \frac{min(R(\mathbf{x}), G(\mathbf{x}), B(\mathbf{x}))}{L_\mathrm{w}(\mathbf{x})} \tag{3.17}$$

the desaturation of each colour channel is achieved by:

$$\begin{bmatrix} R_c(\mathbf{x}) \\ G_c(\mathbf{x}) \\ B_c(\mathbf{x}) \end{bmatrix} = \left(1 - S(\mathbf{x})'\right) \begin{bmatrix} L_\mathrm{w}(\mathbf{x}) \\ L_\mathrm{w}(\mathbf{x}) \\ L_\mathrm{w}(\mathbf{x}) \end{bmatrix} + S(\mathbf{x})' \begin{bmatrix} R_c(\mathbf{x}) \\ G_c(\mathbf{x}) \\ B_c(\mathbf{x}) \end{bmatrix} \qquad S(\mathbf{x})'(\mathbf{x}) = \alpha S(\mathbf{x})^{\beta - 1} \tag{3.18}$$

where $\alpha \leq 1$ is a parameter which controls the level of saturation kept during colour encoding, β is a parameter which determines the colour contrast. After this step, the image is encoded in a modified $\mathrm{YC}_b\mathrm{C}_r$ colour space, because it has a larger gamut than RGB colour space. Therefore, unused $\mathrm{YC}_b\mathrm{C}_r$ values can be exploited to preserve the original gamut of an HDR image. This is achieved by the following mapping:

$$R'(\mathbf{x}) = \begin{cases} 1.055R_c(\mathbf{x})^{0.42} - 0.055 & \text{if } R_c(\mathbf{x}) > 0.0031308 \\ 12.92R_c(\mathbf{x}) & \text{if } |R_c(\mathbf{x})| \leq 0.0031308 \\ -1.055(-R_c(\mathbf{x}))^{0.42} + 0.055 & \text{if } R_c(\mathbf{x}) < -0.0031308 \end{cases} \tag{3.19}$$

which is repeated for the green and blue channel.

The decoding consists of few steps, see Figure 3.15 for the complete pipeline. Firstly, the tone mapped layer is decoded using a JPEG decoder and the gamut is expanded inverting Equation 3.18. After this step, the RI_d image is decoded, expanded (from logarithmic domain to linear domain), and up-sampled to the resolution of the tone mapped layer. Finally, the image is recovered by multiplying the tone mapped layer by the RI_d image.

A first study [213] was conducted to determine a good TMO for compression purposes, which was based on comparison with the original HDR images using VDP [40]. In this experiment different TMOs were compared such as histogram adjustment [107], global photographic tone reproduction operator [169], fast bilateral filtering operator [55], and gradient operator [61]. Experiments showed that fast bilateral filtering operator performed the best followed by the photographic tone reproduction one. A second study was carried out to test image quality and compression rates on a data set of 217 HDR images. The data set was compressed using JPEG-HDR at different quality settings using the global photographic operator, RGBE, OpenEXR and LogLuv TIFF to study compression rates. HDR images compressed using JPEG-HDR were compared with original ones using VDP to study quality of the image. The study showed that the method can achieve a compression rate between 0.6-3.75 bpp for quality settings between $57-99\%$. However, quality degrades rapidly for JPEG quality below 60%, but only 2.5% of pixels were visibly different with a quality set at 90%, and only 0.1% with maximum quality.

Most important, the method is backward compatible, because RI_d is encoded using only extra application markers of JPEG format. When an application not designed for HDR imaging opens a JPEG-HDR file, it displays only the tone mapped layer allowing the user to have access to part of the content.

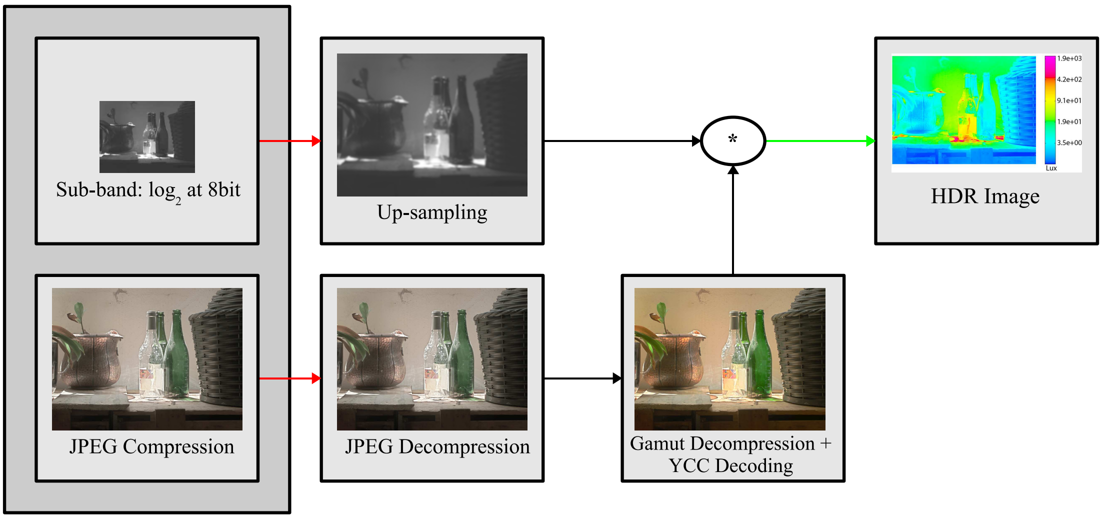


Figure 3.15: *The decoding pipeline for JPEG-HDR by Ward and Simmons [213, 214].*

3.2.2 HDR-JPEG 2000

Xu et al. [223] proposed a simple pre-processing technique which enables the JPEG 2000 standard [33] to encode HDR images. The main concept is to transform floating point data in unsigned short integers (16-bit), that are supported by JPEG 2000 standard.

The encoding phase starts with the reduction of the dynamic range applying the natural logarithm to the RGB values:

$$\begin{bmatrix} R'_{\mathrm{w}}(\mathbf{x}) \\ G'_{\mathrm{w}}(\mathbf{x}) \\ B'_{\mathrm{w}}(\mathbf{x}) \end{bmatrix} = \begin{bmatrix} \log R_{\mathrm{w}}(\mathbf{x}) \\ \log G_{\mathrm{w}}(\mathbf{x}) \\ \log B_{\mathrm{w}}(\mathbf{x}) \end{bmatrix} \tag{3.20}$$

Then, the floating point values in the logarithmic domain are discretised to unsigned short integers:

$$\begin{bmatrix} \overline{R}_{\mathrm{w}}(\mathbf{x}) \\ \overline{G}_{\mathrm{w}}(\mathbf{x}) \\ \overline{B}_{\mathrm{w}}(\mathbf{x}) \end{bmatrix} = \begin{bmatrix} f\left(R'_{\mathrm{w}}(\mathbf{x})\right) \\ f\left(G'_{\mathrm{w}}(\mathbf{x})\right) \\ f\left(B'_{\mathrm{w}}(\mathbf{x})\right) \end{bmatrix} \qquad f(x,n) = (2^n - 1)\frac{x - x_{\mathrm{min}}}{x_{\mathrm{max}} - x_{\mathrm{min}}} \tag{3.21}$$

where x_{max} and x_{min} are respectively the maximum and minimum value for the channel of x, and $n = 16$. Finally, the image is compressed using a classic JPEG 2000 encoder.

The decoding phase is quite straightforward. Firstly, the image is decompressed using a JPEG 2000 decoder, then the integer data is converted into floating point by inverting f in Equation

3.21 which is subsequently exponentiated:

$$\begin{bmatrix} R_{\mathrm{w}}(\mathbf{x}) \\ G_{\mathrm{w}}(\mathbf{x}) \\ B_{\mathrm{w}}(\mathbf{x}) \end{bmatrix} = \begin{bmatrix} e^{g\left(\overline{R}_{\mathrm{w}}(\mathbf{x})\right)} \\ e^{g\left(\overline{G}_{\mathrm{w}}(\mathbf{x})\right)} \\ e^{\left(\overline{B}_{\mathrm{w}}(\mathbf{x})\right)} \end{bmatrix} \qquad g(x,n) = f^{-1}(x,n) = \frac{x}{2^n-1}(x_{\max}-x_{\min})+x_{\min} \tag{3.22}$$

The method was compared in JPEG-2000 lossy mode against JPEG-HDR [214] and HDRV [126], and in JPEG-2000 lossless mode against RGBE [209], LogLuv [106], and OpenEXR [86]. The metrics were RMSE in the logarithm domain and Lubin's VDP [120]. The results of these comparisons showed that HDR JPEG 2000 in lossy mode is superior to JPEG-HDR and HDRV, especially at low bit rates when these methods have artefacts. Nevertheless, the method does not perform well when lossless JPEG 2000 is used, because the file size is higher than RGBE, LogLuv, and OpenEXR (these methods are lossy in the float precision, but not spatially).

The HDR JPEG 2000 algorithm is a straightforward method for lossy compression of HDR images at high quality, without artefacts at low bit rate. However, the method is not suitable for real-time graphics, because fixed time look-ups are needed. Also, the method does not exploit all the compression capabilities of JPEG 2000 because it operates at high level. For example, separate processing for luminance and chromaticity could reduce the size of the final image while keeping the same quality.

3.2.3 Compression and Companding High Dynamic Range Images with Sub-bands Architectures

Li et al. [115] presented a general framework for tone mapping and inverse tone mapping of HDR images based on multi-scale decomposition. While the main goal of the algorithm is tone mapping, in addition, the framework can also compress HDR images. A multi-scale decomposition splits a signal $s(x)$ (1D in this case) into n sub-bands $b_1(x),...,b_n(x)$ with n filters $f_1,...,f_n$, in a way the signal can be reconstructed as:

$$s(x) = \sum_{i=1}^{n} b_i(x) \tag{3.23}$$

Wavelets [186] and Laplacian pyramids [24] are examples of multi-scale decomposition that can be used in the framework.

(a) Activity map (b) Gain map (c) Tone mapped luminance

Figure 3.16: *An example of tone mapping using the multi-scale decomposition with Haar Wavelets. Images are courtesy of Edward Adelson [115].*

The main concept is to apply a gain control to each sub-band of the image to compress the range. For example, a sigmoid expands low values and flats peaks, however it introduces distortions that can appear in the final reconstructed signal. In order to avoid such distortions, a smooth gain map inspired by neurons was proposed. The first step, is to build an activity map, reflecting the fact that the gain of a neuron is controlled by the level of its neighbours. The activity map is defined as:

$$A_i(\mathbf{x}) = G(\sigma_i) \otimes |B_i(\mathbf{x})| \tag{3.24}$$

where $G(\sigma_i)$ is a Gaussian kernel with $\sigma_i = 2^i \sigma_1$ which is proportional to i, the sub-band's scale. The activity map is used to calculate the gain map, which turns gain down where activity is high and viceversa:

$$Gain_i(\mathbf{x}) = p(A_i\mathbf{x}) = \left(\frac{A_i\mathbf{x} + \varepsilon}{\delta_i} \right)^{\gamma - 1} \tag{3.25}$$

where $\gamma \in [0,1]$ is a compression factor, and ε is the noise level that prevents the noise to be seen. $\delta_i = \alpha_i \sum_{\mathbf{x}} A_i(\mathbf{x})/(M)$ is the gain control stability level where M is the number of pixels in the image, and $\alpha_i \in [0.1,1]$ is a constant related to spatial frequency. Once the gain maps are calculated, sub-bands can be modified:

$$B'_i(\mathbf{x}) = Gain_i(\mathbf{x})B_i(\mathbf{x}) \tag{3.26}$$

Note that it is possible to calculate a single activity map for all sub-bands by pooling all activity maps as:

$$A_{ag}(\mathbf{x}) = \sum_{i=1}^{n} A_i(\mathbf{x}) \tag{3.27}$$

From A_{ag}, a single gain map $Gain_{ag} = p(A_{ag})$ is calculated for modifying all sub-bands. The tone mapped image is finally obtained summing all modified sub-bands B'_i, see Figure 3.16. The compression is applied only to the V channel of an image in the HSV colour space [69]. Finally, to avoid over-saturated images S can be reduced by $\alpha \in [0.5,1]$. The authors presented comparisons with the fast bilateral filter operator [55] and gradient domain operator [61], see Figure 3.17.

(a) (b) (c) (d)

Figure 3.17: *A comparison of tone mapping results for the Stanford Memorial Church HDR image: a) The framework using Haar Wavelets. b) The framework using Laplacian Pyramids. c) The fast bilateral filter operator. d) The gradient domain operator. Images are courtesy of Edward Adelson [115].*

The framework can be additionally used for the compression task, applying expansion after tone mapping. This operation is called *companding*. The expansion operation is obtained by a straightforward modification of Equation 3.26:

$$B_i(\mathbf{x}) = \frac{B'_i(\mathbf{x})}{Gain_i(\mathbf{x})} \tag{3.28}$$

A simple companding operation is not sufficient for compression especially if the tone mapped image is compressed using lossy codecs. Therefore, the companding operation needs to be iterative to determine the best values for the gain map, see Figure 3.18. The authors proposed to compress tone mapped image into JPEG. In this case a high bit rate is needed (1.5 bpp - 4bpp) with chrominance sub-sampling disabled to avoid that JPEG artefacts are amplified during expansion, because a simple up-sampling strategy is adopted.

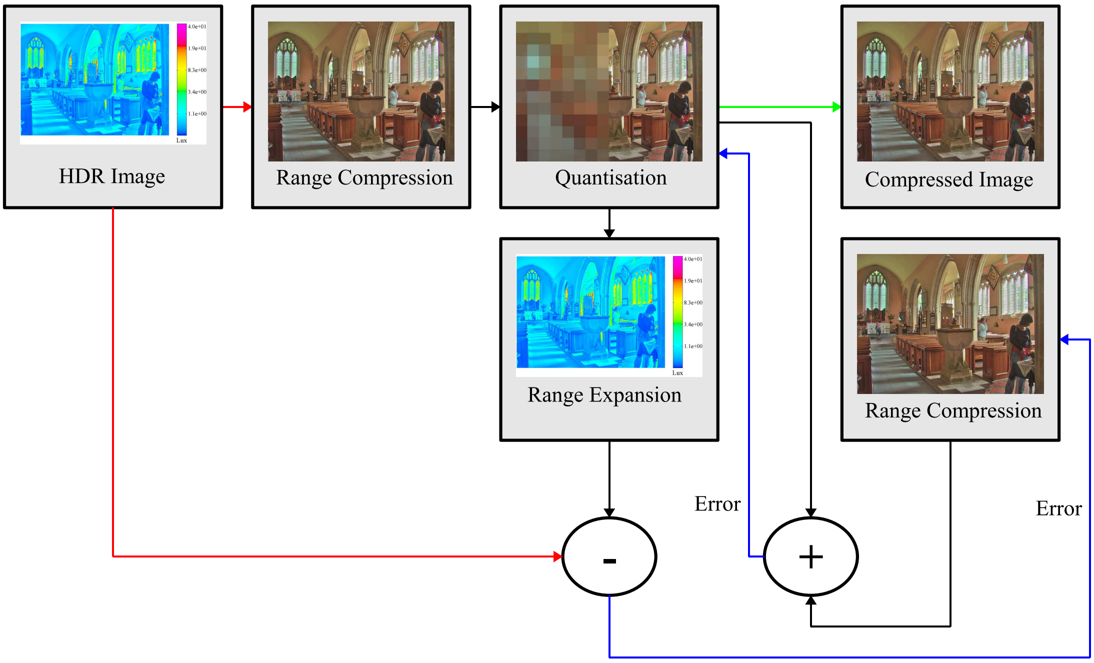


Figure 3.18: *The optimisation companding pipeline of Li et al. [115].*

3.2.4 Backward Compatible HDR-MPEG

Backward compatible HDR-MPEG is a codec for HDR videos that was introduced by Mantiuk et al. [125]. As in the case of JPEG-HDR this algorithm is an extension to the standard MPEG-4 codec (H.264) [220] that works on top of the standard encoding/decoding stage allowing backward compatibility. In a similar way to JPEG-HDR each frame is divided in a LDR part using tone mapping and an HDR part, but in this case for the reconstruction function (RF) a numerical iTMO is employed instead of a RI. HDR-MPEG is an extension to perception motivated video encoding (HDRV) [126], which is not backward compatible.

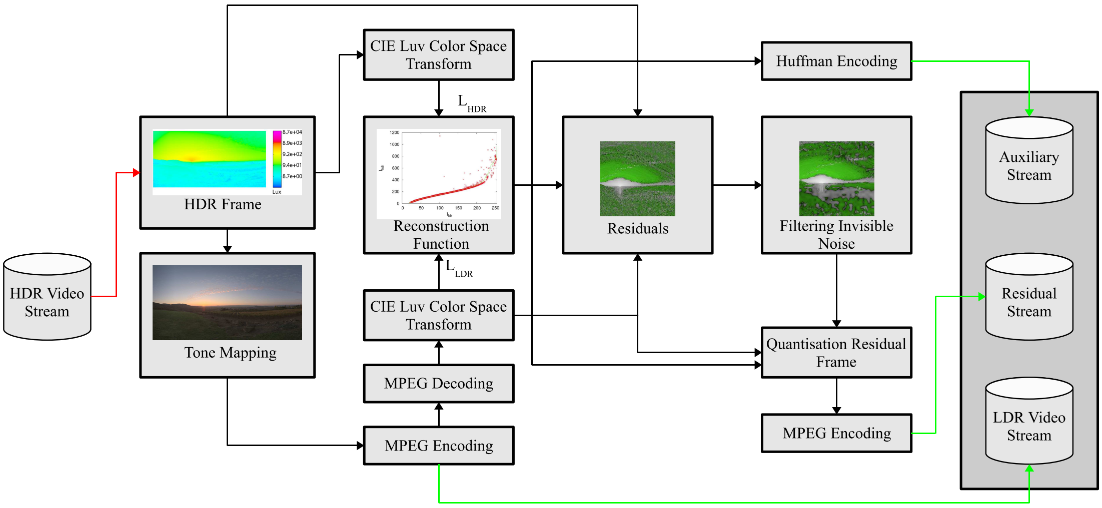


Figure 3.19: *The encoding pipeline for Backward Compatible HDR-MPEG by Mantiuk et al. [125].*

The encoding stage takes as input an HDR video in the XYZ colour space and it applies tone mapping to each HDR frame obtaining LDR frames as first step, see Figure 3.19 for the complete pipeline. These are coded with MPEG-4, stored in an LDR stream, and finally decoded to obtain a uncompressed and MPEG quantised frames. After this, LDR frame and HDR frame are converted to a common colour space. For both HDR and LDR frames CIE 1976 Uniform Chromaticity Scales (u', v') coordinates are used to code chroma. While non-linear luma of sRGB is used for LDR pixels, a different luma coding is used because sRGB non-linearity is not suitable for high luminance ranges $[10^{-5}, 10^{10}]$, see [125]. This luma coding, at 12-bit, for HDR luminance values is given as:

$$l_{\mathrm{w}} = f(L_{\mathrm{w}}) = \begin{cases} 209.16\log(L_{\mathrm{w}}) - 731.28 & \text{if } L_{\mathrm{w}} \geq 10469 \\ 826.81L_{\mathrm{w}}^{0.10013} - 884.17 & \text{if } 5.6046 \leq L_{\mathrm{w}} < 10469 \\ 17.554L_{\mathrm{w}} & \text{if } L_{\mathrm{w}} < 5.6046 \end{cases} \tag{3.29}$$

At this point HDR frames and an LDR ones are in a comparable colour space. Then RF, that maps LDR values, l_{d}, to HDR ones, l_{w}, is calculated in a straightforward way by averaging l_{w} values which fall into one of 256 bins representing the l_{d} values:

$$RF(i) = \frac{1}{|(\Omega_i)|} \sum_{\mathbf{x} \in \Omega_i} l_{\mathrm{w}}(\mathbf{x}) \qquad \text{where } \Omega_i = \left\{ \mathbf{x} | l_{\mathrm{d}}(\mathbf{x}) = i \right\} \tag{3.30}$$

where $i \in [0, 255]$ is the index of a bin. RF for chromaticity is approximated imposing $(u'_{\mathrm{d}}, v'_{\mathrm{d}}) = (u'_{\mathrm{w}}, v'_{\mathrm{w}})$. Once RFs are calculated for all frames, they are stored in a auxiliary stream using Huffman encoding. After this stage, a residual image is calculated for improving overall quality, especially in small details for luma:

$$r_l(\mathbf{x}) = l_{\mathrm{w}}(\mathbf{x}) - RF(l_{\mathrm{d}}(\mathbf{x})) \tag{3.31}$$

The residual image is discretised at 8-bit, using a quantisation factor different for each bin based on its maximum magnitude value, which leads to:

$$\hat{r}_l(\mathbf{x}) = \left[\frac{r_l(\mathbf{x})}{q(m)}\right]_{-127}^{127} \qquad \text{where } m = k \Leftrightarrow i \subset \Omega_k \tag{3.32}$$

where $q(m)$ is the quantisation factor which is calculated for a bin Ω_l as:

$$q(m) = max\left(q_{min}, \frac{max_{\mathbf{x}\in\Omega_l}(|r_l(\mathbf{x})|)}{127}\right) \tag{3.33}$$

Then $\hat{r}_l$ needs to be compressed in a stream using MPEG, but a naïve compression would generate a low compression rate, because a large amount of high frequencies is present in $\hat{r}_l$. In order to improve the compression rate, the image is filtered removing frequencies in regions where the HVS can not perceive any difference. This is achieved using the original HDR frame as a guidance in the filtering. The filtering is performed in the wavelet domain, and it is applied only to the three finest scales modelling contrast masking, and lower sensibility to high frequencies.

The decoding stage is quite straightforward. MPEG streams (tone mapped video and residuals) and RF streams are decoded, see Figure 3.20 for the complete pipeline. Then, an HDR frame is reconstructed applying firstly its RF to the LDR decoded frame, and secondly adding residuals to the expanded LDR frame. Finally, CIE Luv values are converted to XYZ ones inverting Equation 3.29 for luminace.

HDR-MPEG was evaluated using three different metrics: HDR VDP [127, 124], universal image quality index (UQI) [208], and classic Signal to Noise Ratio (SNR). As in the case of JPEG-HDR, there was a first study that explored the influence of a TMO on quality/bit rate.

This experiment was performed using different TMOs such as time dependent visual adaption [156], fast bilateral filtering [55], photographic tone reproduction [169], gradient domain [61] and adaptive logarithmic mapping [52]. These TMOs were modified for avoiding temporal flickering and applied to a stream using default parameters. The conclusion of the study showed that most of these TMOs have the same performances except the gradient domain one, which creates larger streams than others. However, this TMO generated more attractive images for backward compatibility, therefore the choice of a TMO for the video compression depends by the trade-off between bit rate and the backward compatible quality. The second study compared HDR-MPEG against HDRV [126] and JPEG-HDR [214] using the photographic tone reproduction operator as the TMO [169]. The results showed that HDR-MPEG has a better quality than JPEG-HDR, but a similar one to HDRV.

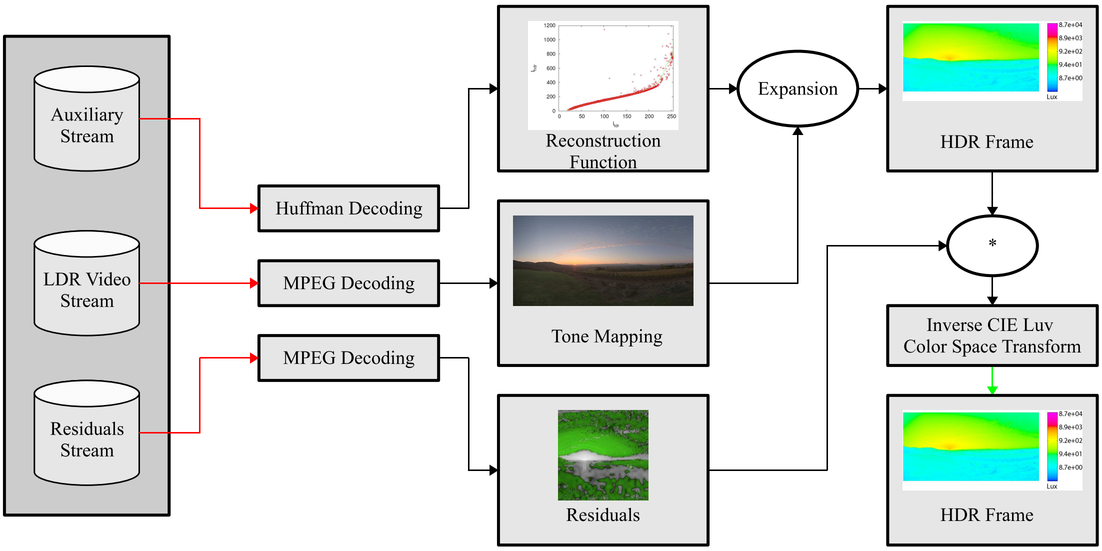


Figure 3.20: *The decoding pipeline for Backward Compatible HDR-MPEG by Mantiuk et al. [125].*

3.2.5 Encoding of High Dynamic Range Video with a Model of Human Cones

Similarly to Li et al. [115], Van Hateren [75] proposed a new tone mapping operator based on a model of human cones [76] which can be inverted to encode HDR images and videos. The TMO and iTMO work in troland units (td), the measure of retinal illuminance I, which is derived by the scene luminance in cd/m^2 multiplied by the pupil area in mm^2. Van Hateren proposed a temporal and a static version of its TMO.

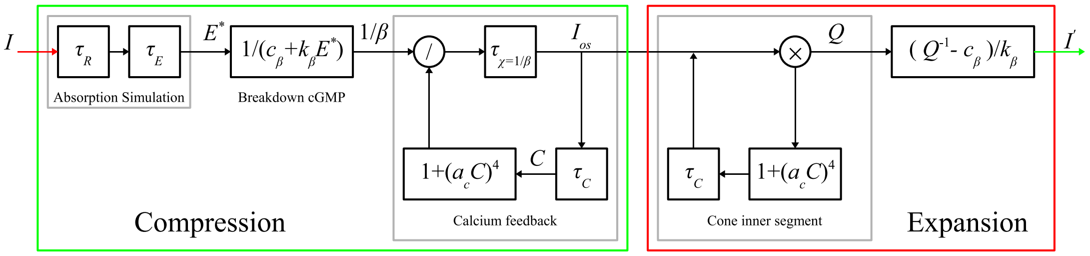


Figure 3.21: *The pipeline for range compression (red) and range expansion (green) proposed by Van Hateren [75].*

The temporal TMO is designed for HDR videos and presents low-pass temporal filters for removing photon and source noise, see Figure 3.21. The TMO starts by simulating the absorption of I by visual pigment, which is modelled by two low-pass temporal filters and described in terms of a differential equation:

$$\frac{dy}{dt} + \frac{1}{\tau} y = \frac{1}{\tau} x \tag{3.34}$$

where τ is a time constant, $x(t)$ is the input at time t, and $y(t)$ is the output. At this point, a strong non linearity is applied to the result of low-pass filters E^* for simulating the breakdown of cyclic guanosine monophosphate (cGMP) by enzymes (cGMP is a nucleotide, that controls the current across the cell membranes):

$$\alpha = \frac{1}{\beta} = \left(c_\beta + k_\beta E^* \right)^{-1} \tag{3.35}$$

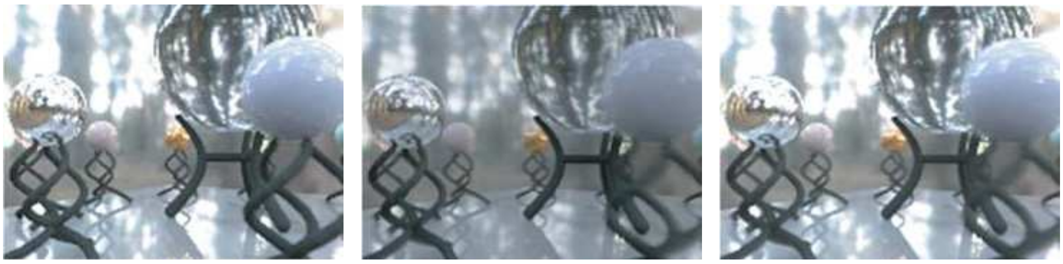
where $k_\beta E^*$ is the light-dependent activity of an enzyme, and c_β the residual activity. The breakdown of cGMP is counteracted by the production of cGMP; a highly non-linear feedback loop under control of inter-cellular calcium. This system is modelled by a filtering loop which outputs the current across cell membrane, I_{os} (the final tone mapped value), by the outer segment of a cone.

Van Hateren showed that the range expansion is quite straightforward by inverting the feedback loop. However, the process can not be fully inverted because the first two low-pass filters are difficult to invert, so the result is $I' \approx I$. In order to fully invert the process for inverse tone mapping purposes Van Hateren proposed a steady version of the TMO, a global TMO, defined as:

$$I_{\text{os}} = \frac{1}{(1+(a_C I_{\text{os}})^4)(c_\beta + k_\beta I)} \tag{3.36}$$

which can be easily inverted as:

$$I = \frac{1}{k_\beta}\left[\frac{1}{I_{\text{os}}(1+(a_C I_{\text{os}})^4)} - c_\beta\right] \tag{3.37}$$



(a) (b) (c)

Figure 3.22: *An example of Van Hareten's algorithm with a frame of the RNL sequence: a) original f-stop 0. b) Tone mapped frame using cone model. c) Reconstructed frame using proposed iTMO. Images are courtesy of Van Hareten [75].*

Van Hateren applied its TMO and iTMO to uncalibrated HDR movies and images, which were scaled by harmonic mean. His results showed that the method does not need gamma correction, removes noise, and present light adaptation, see Figure 3.22. The main drawbacks of the TMO are the introduction of motion blur in movies, and the limited dynamic range that can handle, 10,000:1, which causes saturation regions in very dark and bright regions. Finally, the author does not provide any study on companding and further quantisation for his TMO/iTMO.

3.2.6 Two-layer Coding Algorithm for High Dynamic Range Images Based on Luminance Compensation

A similar compression scheme to HDR-MPEG was proposed by Okuda and Adami [146] for compression of HDR images. This scheme is designed to be backward compatible as JPEG-HDR and HDR-MPEG. The main differences are the presence of a minimisation step for optimising tone mapping parameters, the compression of residuals using wavelets, and the use of the Hill function for tone mapping and its analytic inverse instead of a tabled function as in HDR-MPEG. The Hill function is a generalised sigmoid function which is defined as:

$$L_{\mathrm{d}}(\mathbf{x}) = f(L_{\mathrm{w}}(\mathbf{x})) = \frac{L_{\mathrm{w}}(\mathbf{x})^n}{L_{\mathrm{w}}(\mathbf{x})^n + k^n} \tag{3.38}$$

where n and k are a constants that depend on the image. The inverse g of f is given by:

$$L_{\mathrm{w}}(\mathbf{x}) = g(L_{\mathrm{d}}(\mathbf{x})) = f^{-1}(L_{\mathrm{d}}(\mathbf{x})) = k\left(\frac{L_{\mathrm{d}}(\mathbf{x})}{1 - L_{\mathrm{d}}(\mathbf{x})}\right)^{\frac{1}{n}} \tag{3.39}$$

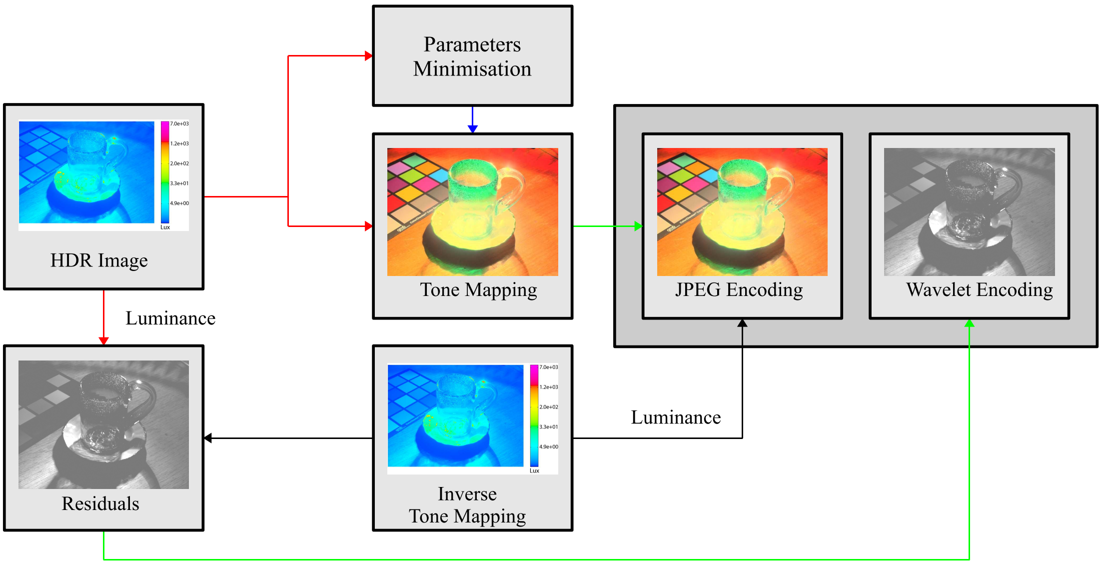


Figure 3.23: *The encoding pipeline presented of Okuda and Adam's method [146].*

The encoding is divided into a few steps, see Figure 3.23. Firstly, a minimisation process using the original HDR image is performed in the logarithmic domain to match HVS perception and avoid outliers at high values. The error to minimise is given by:

$$E = \sum_{\mathbf{x}\in I}\Big(\log(L_{\mathrm{w}}(\mathbf{x})) - \log(g(L_{\mathrm{d}}(\mathbf{x})))\Big)^2 \tag{3.40}$$

for determining n and k. The optimum solution is uniquely determined imposing the partial derivatives of E for k and n equal to zero, leading to:

$$\begin{aligned} k &= \exp\left(\frac{\left(\sum_{\mathbf{x}} B(\mathbf{x})^2\right)\left(\sum_{\mathbf{x}} A(\mathbf{x})\right) - \left(\sum_{\mathbf{x}} B(\mathbf{x})\right)\left(\sum_{\mathbf{x}} A(\mathbf{x})B(\mathbf{x})\right)}{M\left(\sum_{\mathbf{x}} B(\mathbf{x})^2\right) - \left(\sum_{\mathbf{x}} B(\mathbf{x})^2\right)}\right) \\ n &= \frac{M\left(\sum_{\mathbf{x}} B(\mathbf{x})^2\right) - \left(\sum_{\mathbf{x}} B(\mathbf{x})\right)^2}{M\left(\sum_{\mathbf{x}} A(\mathbf{x})B(\mathbf{x})\right) - \left(\sum_{\mathbf{x}} A(\mathbf{x})\right)\left(\sum_{\mathbf{x}} B(\mathbf{x})\right)} \end{aligned} \tag{3.41}$$

where M is the number of pixels, and A and B are defined as:

$$A(\mathbf{x}) = \log L_{\mathrm{w}}(\mathbf{x}) \qquad B(\mathbf{x}) = \log\left(\frac{L_{\mathrm{d}}(\mathbf{x})}{1 - L_{\mathrm{d}}(\mathbf{x})}\right) \tag{3.42}$$

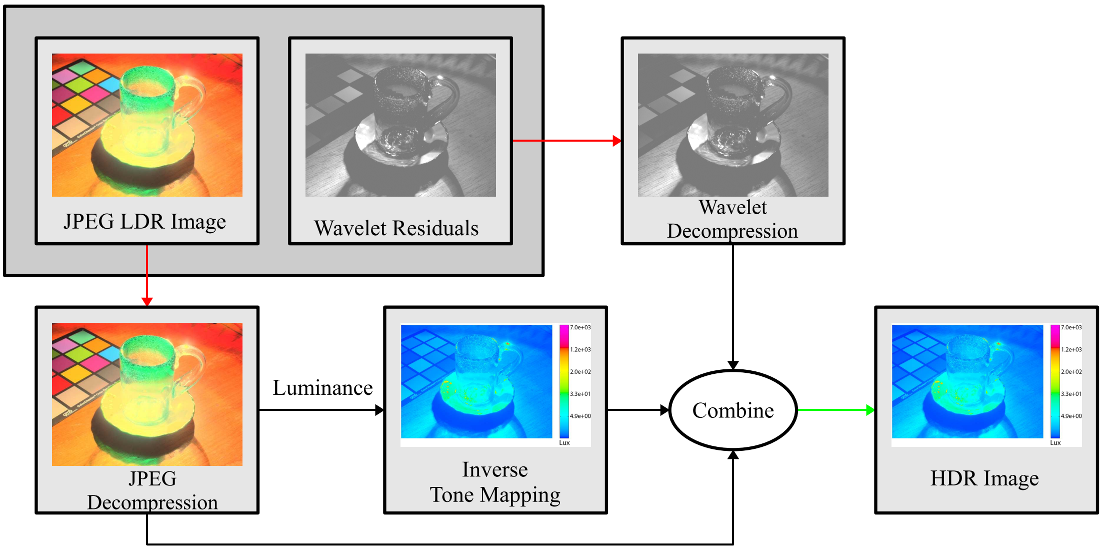


Figure 3.24: *The decoding pipeline presented of Okuda and Adami's method [146].*

Once the parameters are determined the image is tone mapped and encoded using JPEG. To improve quality, residual are calculated as:

$$R(\mathbf{x}) = \left(\frac{L_{\mathrm{w}}(\mathbf{x})}{g(L_{\mathrm{d}}(\mathbf{x})) + \varepsilon}\right)^{\gamma} \tag{3.43}$$

where $\gamma \in (0, 1]$ is a constant, and $\varepsilon \geq 0$ is a small value to avoid discontinuities chosen by the user. Finally, R is encoded using a wavelet image compression scheme.

The decoding stage is straightforward. Once the LDR image and residuals are decoded using a JPEG decoder and a wavelet decoder, final HDR values are recovered by:

$$L_{\mathrm{w}}(\mathbf{x}) = R(\mathbf{x})^{\frac{1}{\gamma}} \left(g(L_{\mathrm{d}}(\mathbf{x})) + \varepsilon\right) \tag{3.44}$$

Two colour compensation methods are presented to preserve distortions caused by tone mapping. The first one is a modification of Ward and Simmons [214] where α and β are calculated with a quadratic minimisation using an error function similar to Equation 3.40. The second method is to apply a polynomial $P(x)$ for each LDR colour channel, assuming that a polynomial relationship exists between LDR and HDR values. Coefficients of $P(x)$ are fitted using the Gaussian weighted difference between the original HDR channel and the reconstructed HDR channel.

The compression scheme was evaluated on a data set of 12 HDR images and compared with JPEG-HDR and HDR-MPEG using two metrics: the mean square difference (MSE) in CIELAB colour space [58] and MSE in the Daly's non-linearity domain [40]. In these experiments the proposed method achieved better results for both metrics in comparison with JPEG-HDR and HDR-MPEG at different bit rates. While the quality of this method is up to two times better than HDR-MPEG and JPEG-HDR at high bit rates (around 8-10 bits), it is comparable to them at low bit rates (around 1-4 bits).

3.3 Summary

In this chapter the state of art of inverse tone mapping was reviewed. The first part described methods for expanding single exposure LDR content into HDR content that can be employed for HDR applications. The second part showed compression techniques that exploit tone mapping, inverse tone mapping, and de facto compression standards for efficient compression of HDR images and videos.

Chapter 4

Methodology

This chapter presents the methodology that was adopted for the developing of the new algorithm for the expansion of LDR content into HDR content. This is presented in Chapter 5. The methodology follows an iterative process, and all iterations are presented as well.

4.1 The Method

Inverse tone mapping bridges the gap between LDR and HDR imaging with very practical applications as shown in previous chapters. Therefore, a practical approach was employed in the design of the final solution in Chapter 5. The adopted methodology can be seen as a simplified version of Action Research methodology. The starting of the process is an existing algorithm for expanding LDR content into HDR one. The algorithm is modified at each iteration until the wanted result is reached. At each iteration a set of images is expanded and used for an applications (IBL or image visualisation). Then they are analysed to understand which features are important, and which lead to drawbacks. Finally, the algorithm is modified in accord with the analysed data. If there are no more modifications, the process stops, otherwise another iteration is performed. The diagram of this process is shown in Figure 4.1.

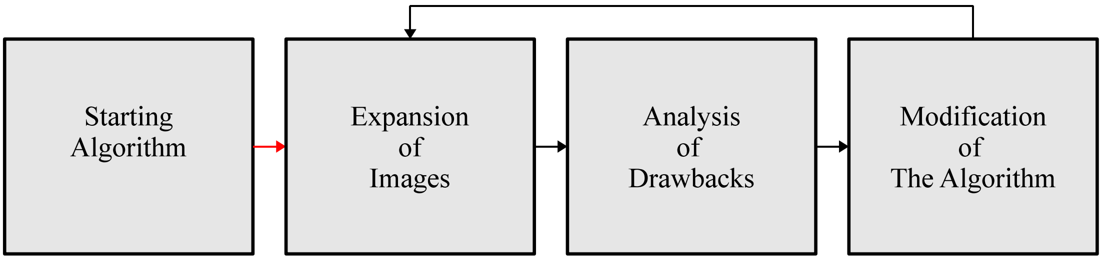


Figure 4.1: *The diagram of the used methodology. A starting algorithm is used for expanding images. Then images are analysed to understand which are the important features and drawbacks. Afterwards the algorithm is modified to add or remove some features. If no modifications are added the goal algorithm is obtained otherwise the process is reiterated.*

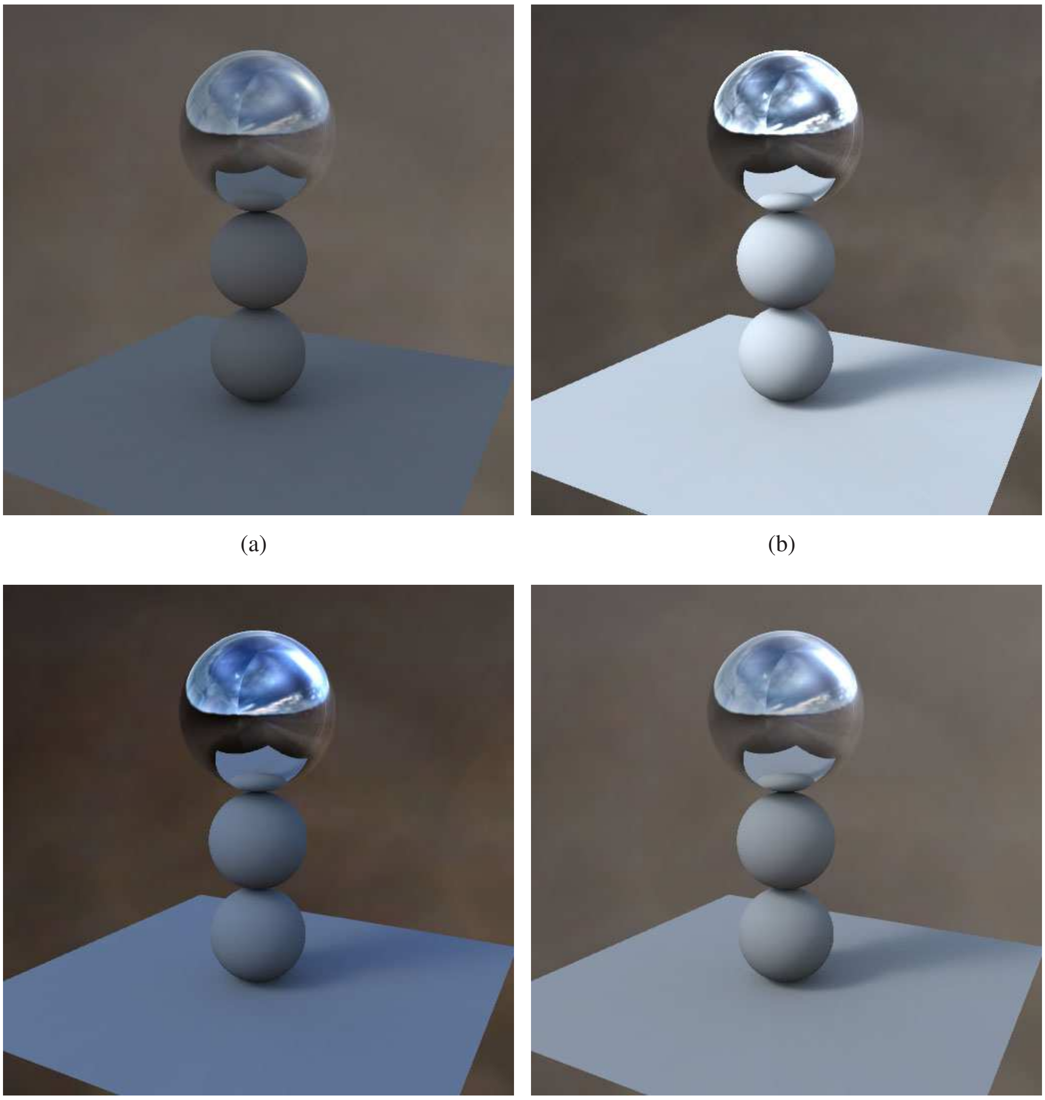
4.2 Starting Solutions

At the time of developing of the main algorithm there were only a few solutions for expanding LDR content into HDR one. Landis' method [105] and Daly and Feng's ones [41, 42] were the only available algorithms. Moreover, they model a specific application and not a general case. For example, while Landis' operator can expand LDR images which are suitable for IBL, it produces images with noticeable contouring artifacts. Daly and Feng's algorithm removes contours for image visualisation, but the generated images do not enhance light sources. This means that it performs very poorly for IBL applications. Between these two algorithms, Landis's one was chosen as starting algorithm, because IBL is a very important application in computer graphics, visual effects, videogames, and automotive industry.

4.3 The Iteration of the Process

The Landis operator expands pixel values over a certain threshold using a power function. Figure 4.2.b shows that the operator produces a better result than a linear one, Figure 4.2.a, regarding high frequency shadows. This can produce visible artifacts such as contours in the environment map. These can be noticed if the environment map is visualised when highly specular materials are employed in the virtual scene.

In the first iteration, the thresholding was removed for understanding if it is needed in the expansion, for an example see Figure 4.2.c. Note that the high contrast contour is not present anymore in the reflections of the specular sphere. However, the control of the user is limited



(a) (b)

(c) (d)

Figure 4.2: *An example of the iterations for IBL: a) A render using an LDR environment map. b) Landis's method [105]. c) Landis's method without thresholding. d) Expansion through an inverse TMO and expand map.*

to the power coefficient, not allowing to focus the enhancement only on high luminance areas. Furthermore, colours are greatly modified increasing saturation due to the power function application.

In order to expand only certain pixel values, as in thresholding but avoiding its main disadvantage, the power function was replaced with the inverse of a TMO. This is because the inverse of a TMO typically expands high luminance values, keeps medium values, and compresses low values. However, light sources were not properly reconstructed, they were flat producing only hard shadows. Therefore, a reconstruction mechanism was added, the Expand Map, for an example see Figure 4.2.d. Moreover, the expand map can reduce quantisation artifacts due to expansion at the edges between high and medium/low luminance areas.

A final iteration was performed for extending the algorithm to videos in order to avoid temporal flickering, colour correction, and colour reconstruction. This can happen when frames are separately processed.

4.4 Summary

In this chapter, the adopted methodology is reviewed. Furthermore, the main iterations of the process are shown, explaining the choices made and how they related to learning from previous literature.

Chapter 5

An Inverse Tone Mapping Operator

This chapter describes an iTMO [16, 18, 17] which can expand LDR images for visualisation on HDR displays and IBL.

In Section 5.1 the core concepts of the algorithm for static images are presented. In Section 5.2 these are extended to video sequences where all parameters are automatically estimated and flickering is minimised. Finally, an efficient implementation that exploits graphics hardware and multi-core CPU is exposed in Section 5.3.

5.1 Inverse Tone Mapping for Still Images

The developed iTMO was firstly designed for static images which can be generalised into a framework. This is outlined in Figure 5.1, and it comprises of the following steps:

1. **Linearisation**: the starting LDR image is linearised, to produce controllable and consistent results as discussed in Section 3.1.1. The users can choose different methods of linearisation based on the source of the image.
2. **Range Expansion**: an initial HDR image is generated by applying an expansion function to the LDR image generated at Step 1. Many monotonically increasing TMOs can be used for this purpose by using the inverted form.

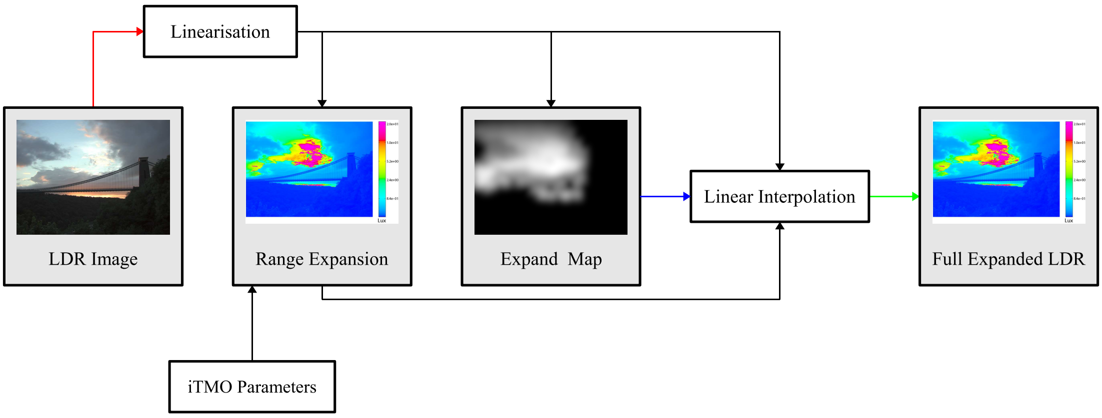


Figure 5.1: *The pipeline of the framework for the iTMO for still images. The LDR image is expanded using an expansion function obtaining a preliminary HDR image. Then, an expand map is generated from the original LDR image by calculating the density estimation of light source samples. Finally, the original LDR and the preliminary HDR image are interpolated using the expand map as interpolation weight.*

3. **Expand Map**: a smooth field is generated to reconstruct clipped areas and reduce noise. This field is called *expand map* and it is created from density estimation of light source samples generated using importance sampling.
4. **Combination**: The final HDR image is generated by interpolating the original LDR image and the expanded one at Step 2 using the expand map as interpolation weights. This process reduces artefacts and reconstructs clipped areas.

5.1.1 Linearisation of the Signal

Linearisation is an important step for an iTMO, because it allows to control the results of the expansion, see Section 3.1.1. In the developed system there are three options for linearising LDR images: known cameras, unknown cameras, and DVDs or television programs. The selection of an option depends by the knowledge of the user about the device that captured LDR images as input.

For the first option, the user needs to determine the CRF of this camera, see Section 2.2.2, and to apply the inverse CRF to the image to obtain the linearised image. For the second option, algorithms for linearisation from a single image are applied, see Section 3.1.1. Finally, in the last option, inverse gamma of 2.2 is applied to an image, because DVDs and television programs formats encode videos using this curve as described in the ITU-R BT.709 standard

[89].

Note that the presence of three options allows to have the best precise results, because the blind application of single image linearisation algorithms such as Lin et al. [116] and Lin and Zhang [117] can present errors in the estimation of the CRF.

5.1.2 Pixel Values Expansion

A key part of the framework is the enhancement of the original signal, in particular how pixels are expanded to obtain HDR values. The expansion is driven by a monotonically increasing function g assuming the hypothesis that the order of luminance values is kept in the LDR image, which means:

$$\forall \mathbf{x},\mathbf{y} \quad L_{\mathrm{d}}(\mathbf{x}) \leq L_{\mathrm{d}}(\mathbf{y}) \leftrightarrow L_{\mathrm{w}}(\mathbf{x}) \leq L_{\mathrm{w}}(\mathbf{y}) \tag{5.1}$$

where L_{d} is the luminance channel of the LDR image, and L_{w} would be the luminance channel of an HDR image if it was taken of the same scene and position of the LDR one. g is defined as:

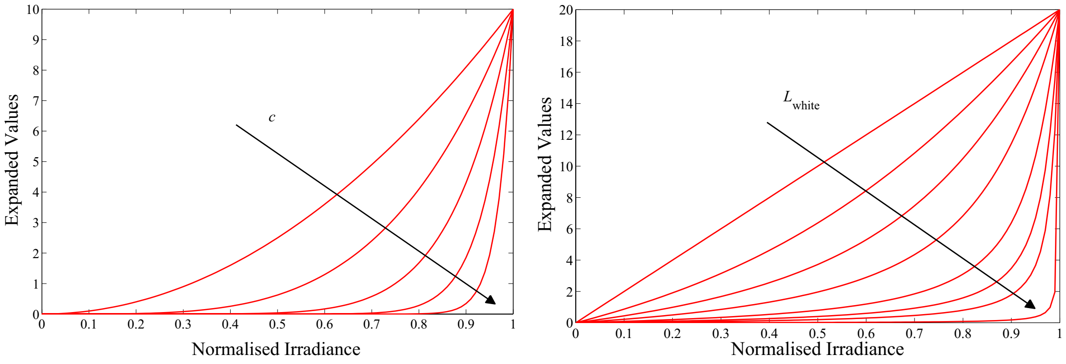
$$g(I) : [0,255]^{w\times h\times c} \rightarrow \mathbb{R}_i^{w\times h\times c} \tag{5.2}$$

where I is the input LDR image, w, h, and c are respectively the width, the height, and the number of colour channels of I. While g tries to expand over-exposed pixels and compress under-exposed ones, it keeps linearly well-exposed ones similarly to a TMO. This behaviour is the opposite of a global TMO. However, it is convenient to work only on the luminance channel to avoid colour shifts during expansion, a similar problem that happens in tone mapping. This leads Equation 5.2 to:

$$g(I) = \begin{cases} L_{\mathrm{w}} = g_{\mathrm{L}}(L_{\mathrm{d}}) : [0,255]^{w\times h} \rightarrow \mathbb{R}_i^{w\times h} \\ \\ \begin{bmatrix} R_{\mathrm{w}} \\ G_{\mathrm{w}} \\ B_{\mathrm{w}} \end{bmatrix} = \frac{L_{\mathrm{w}}}{L_{\mathrm{d}}} \begin{bmatrix} R_{\mathrm{d}} \\ G_{\mathrm{d}} \\ B_{\mathrm{d}} \end{bmatrix} \end{cases} \tag{5.3}$$

A simple approach to define g is to invert an existing TMO, because just an inverse behaviour is needed as it was shown before. This approach was adopted focusing on inverting a non linear global TMO with a small number of parameters. Non linearity is important as in the case of tone mapping, Schlick [178] addressed the fact that a linear mapping produces a not well distributed quantisation, producing very dark images in many cases. Furthermore a linear mapping produces quantisation problems in the case of inverse tone mapping. This appears in the form of contouring as shown in Daly and Feng's work [41, 42].

A small number of parameters is crucial to help users make and to allow an automatic estimate for video sequences. However, good control to users in how the expansion has to be led needs to be kept. Finally, the operator has to be computationally efficient for real-time applications such as streaming live events or movies playback.



(a)

(b)

Figure 5.2: *iTMO curves: a) The inverse of power function. In this example different curves are plotted with increasing value for c: 2, 4, 8, 16, and 32. b) The inverse of the photographic tone reproduction operator, showing how L_{white} is related to the shape of the iTMO. In this example different curves are plotted with increasing value for L_{white}: 1, 2.5, 5, 10, 25, 50, 100, and 1000 (the arrow shows the increasing direction).*

Only a few global operators have all these features and they are: non uniform quantisation [178], global photographic tone reproduction [169], time dependent visual adaptation [156], and more in general sigmoid functions. Other popular TMOs are not suitable for being inverted for expansion purposes. For example, linear scaling operators [210, 63] do not increase the dynamic range. Furthermore, the inverted revised Tumblin-Rushmeier [193], which represents the family of power functions $L_{\text{w}}(\mathbf{x}) = L_{\text{w, Max}} L_{\text{d}}(\mathbf{x})^c$, does not provide a good control of the shape. For the same reason logarithmic and exponential functions are not applicable to this problem. Moreover, medium and high values for c produce an extreme flat profile for mid and low LDR values, see Figure 5.2.a, which can create false contours.

Among the three suitable operators, which are all sigmoid functions, it was decided to invert the photographic tone reproduction operator. For example, the time dependent visual adaptation [156] was discarded because it is excessively computationally complex. Furthermore, inverted sigmoids of cones and rods are not suitable for a single exposure because they are not related to perceptual quantities in the general case.

For clarity the formula of the photographic tone reproduction operator is recalled from Section 2.4.2:

$$L_{\mathrm{d}}(\mathbf{x}) = \frac{\alpha L_{\mathrm{w}}(\mathbf{x})\left(\alpha L_{\mathrm{w}}(\mathbf{x}) + L_{\mathrm{w,\,H}} L_{\mathrm{white}}^2\right)}{L_{\mathrm{w,\,H}} L_{\mathrm{white}}^2 \left(\alpha L_{\mathrm{w}}(\mathbf{x}) + L_{\mathrm{w,\,H}}\right)} \tag{5.4}$$

This equation can be straightforwardly inverted. The first step is to group all terms on the left side obtaining:

$$\frac{\alpha^2}{L_{\mathrm{white}}^2 L_{\mathrm{w,\,H}}^2} L_{\mathrm{w}}(\mathbf{x})^2 + \frac{\alpha}{L_{\mathrm{w,\,H}}}\left(1 - L_{\mathrm{d}}(\mathbf{x})\right) L_{\mathrm{w}}(\mathbf{x}) - L_{\mathrm{d}}(\mathbf{x}) = 0 \tag{5.5}$$

The full inversion is achieved by finding the roots of second order polynomial in $L_{\mathrm{w}}(\mathbf{x})$, which leads to the final iTMO:

$$L_{\mathrm{w}}(\mathbf{x}) = g\mathrm{L}(L_{\mathrm{d}}(\mathbf{x})) = \frac{L_{\mathrm{white}}^2 L_{\mathrm{w,\,H}}}{\alpha}\left(L_{\mathrm{d}}(\mathbf{x}) - 1 + \sqrt{\left(1 - L_{\mathrm{d}}(\mathbf{x})\right)^2 + \frac{4}{L_{\mathrm{white}}^2} L_{\mathrm{d}}(\mathbf{x})}\right) \tag{5.6}$$

Note that Equation 5.5 can have two solutions, but the positive one is chosen because the iTMO has to generate positive values. Moreover, Equation 5.6 can be used for expansion of pixels, and it is always positive for $\alpha > 0$, and $L_{\mathrm{d}}(\mathbf{x}) \geq 0$.

The only parameters that users need to specify are α, $L_{\mathrm{w,\,H}}$, and L_{white}. The first two parameters can be combined together in a single one $\beta = \frac{L_{\mathrm{w,\,H}}}{\alpha}$, because they appear both in the single place as multiplicative coefficients of Equation 5.6. This simplifies users task for parameter choice, obtaining a new equation:

$$L_{\mathrm{w}}(\mathbf{x}) = g_{\mathrm{L}}(L_{\mathrm{d}}(\mathbf{x})) = L_{\mathrm{white}}^2 \beta \left(L_{\mathrm{d}}(\mathbf{x}) - 1 + \sqrt{\left(1 - L_{\mathrm{d}}(\mathbf{x})\right)^2 + \frac{4}{L_{\mathrm{white}}^2} L_{\mathrm{d}}(\mathbf{x})}\right) \tag{5.7}$$

$\beta \in (1, +\infty)$ and $L_{\text{white}} \in (1, +\infty)$ map the range from $[0, 1]$ to a subset of $[0, +\infty)$. However, it is not intuitive for an user to determine the desired scale from two parameters. To simplify this task, an interface parameter is added, which determines the maximum expansion value, called $L_{\text{w, Max}}$, for the maximum value in the LDR image, $L_{\text{d, Max}}$. β is calculated from $L_{\text{w, Max}}$ by solving Equation 5.7 for β with $L_{\text{d}}(\mathbf{x}) = L_{\text{d, Max}}$ and $L_{\text{w}}(\mathbf{x}) = L_{\text{w, Max}}$, obtaining:

$$\beta = \frac{L_{\text{w, Max}}}{L^2_{\text{white}}\left(L_{\text{d, Max}} - 1 + \sqrt{\left(1 - L_{\text{d, Max}}\right)^2 + \frac{4}{L^2_{\text{white}}} L_{\text{d, Max}}}\right)} \tag{5.8}$$

Note that $L_{\text{d, Max}}$ can be assumed to be equal to 1 which is the maximum LDR value, so Equation 5.8 can be simplified as:

$$\beta = \frac{L_{\text{w, Max}}}{2L_{\text{white}}} \tag{5.9}$$

This speeds up the expansion, because $L_{\text{d, Max}}$ does not need to be calculated. The other parameter, L_{white}, determines the shape of the curve, and it is directly proportional to curve steepness, see Figure 5.2.b for an empirical study of the curve increasing L_{white}. Note that the steepness is increased significantly using multiples of the order of $L_{\text{w, Max}}$.

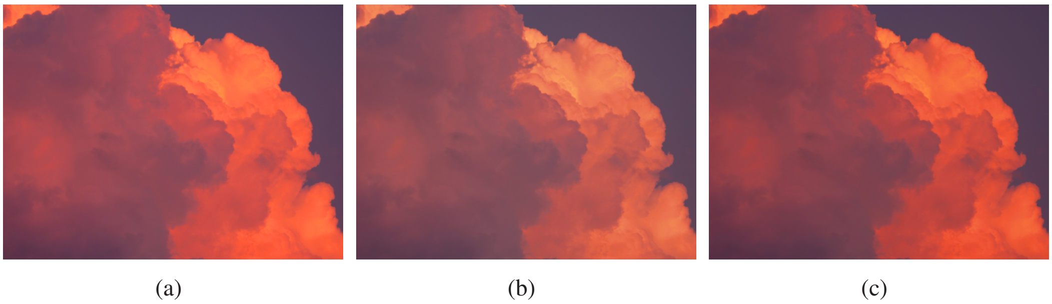

(a) (b) (c)

Figure 5.3: *An example of saturation recovery using Equation 5.11 applied to the Cloud LDR image: a) A zoom of the original LDR image. b) The expanded image in a) without saturation recovery at a visualisable exposure. c) The image in b) with saturation recovery. Note that red colours are more saturated in a) and c) compared to b).*

The main problem of expansion is that colours can be desaturated. This is the contrary effect of a typical TMO, where colours are over-saturated, see Figure 5.3.b for an example. This problem is solved in a straightforward way by increasing the level of saturation:

$$\begin{bmatrix} R_{\mathrm{w}} \\ G_{\mathrm{w}} \\ B_{\mathrm{w}} \end{bmatrix} = L_{\mathrm{w}} \left(\frac{1}{L_{\mathrm{d}}} \begin{bmatrix} R_{\mathrm{d}} \\ G_{\mathrm{d}} \\ B_{\mathrm{d}} \end{bmatrix} \right)^{s} \qquad s = h(L_{\mathrm{w}}, L_{\mathrm{d}}) \geq 1 \tag{5.10}$$

where h is a function that estimates the needed saturation based on the amount of expansion. Its definition is related to g for guiding saturation and avoiding over-saturation. The concept is to assign a high saturation values to expanded values, and to keep saturation for unexpanded ones. A possible h that satisfies these conditions can be defined as:

$$h(\mathbf{x}) = S_{\mathrm{Max}}(1 - 3t(\mathbf{x})^2 + 2t(\mathbf{x})^3) + S_{\mathrm{Min}}(3t(\mathbf{x})^2 - 2t(\mathbf{x})^3) \qquad t(\mathbf{x}) = \frac{L_{\mathrm{d}}(\mathbf{x})}{L_{\mathrm{w}}(\mathbf{x})} \tag{5.11}$$

where S_{Max} and S_{min} are respectively the maximum and minimum saturation value which are interpolated using a cubic with ratio r_{L} as weight. Note that $S_{\mathrm{Min}} = 1$, because unexpanded values does not need to be corrected. From experiments, it was found $S_{\mathrm{Max}} = 2$ to avoid exaggeration in the final saturation. An example of this technique can be seen in Figure 5.3.

5.1.3 Expand Map

A point-wise range expansion can be sufficient in the case of images with well-exposed pixels, such as an outdoor picture in a cloudy day, see Figure 5.4 for an example.

(a) (b)

Figure 5.4: *An example of expansion for well-exposed content for the Vineyard LDR image: a) The original LDR image. b) The false colour image of the expanded image in a) using Equation 5.7 with* $L_{w,\,Max} = 20$ *and* $L_{white} = 40$. *Note that there are no over/under-exposed, so there are no missing areas with details and colours.*

However, more sophisticated image processing is needed when saturated areas are in the image, such as very bright sky in a sunny outdoor picture or a lamp in an indoor scene, see Figure 5.5

for an example. Indeed, saturated regions are clipped with no extra information about the original signal, and an expansion can not generate the lost content.

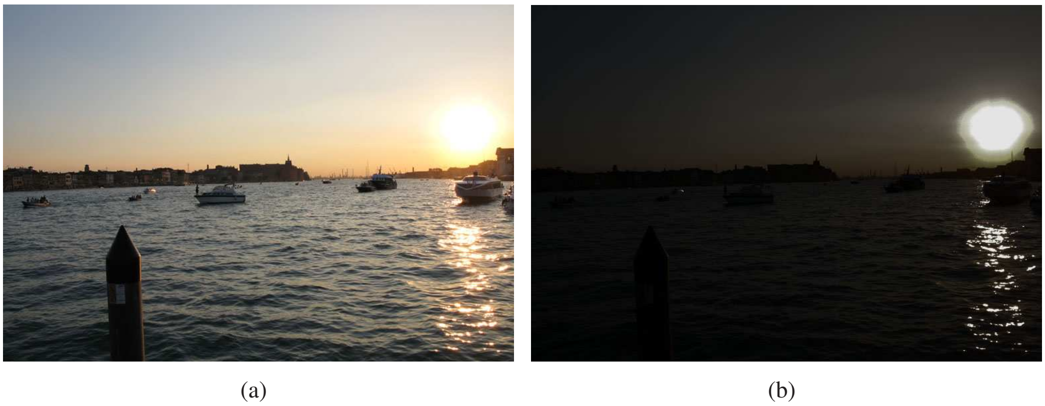

(a) (b)

Figure 5.5: *An example of expansion for over-exposed content for the Venice Bay LDR image: a) The original LDR image. b) F-stop -4 for the expanded version using Equation 5.7 with* $L_{w,\, Max} = 40$ *and* $L_{white} = 80$*. Note that quantisation artefacts are around the sun and there is no reconstruction of content, and the sun is uniformly white.*

Assuming that over-exposed pixels belong to light sources, or self-emitting or reflecting surfaces, a solution is to reconstruct the low frequency profiles of these parts of the scenes using neighbours of saturated pixels to retrieve information about the original signal. Moreover, a large neighbourhood needs to be used in the case of a large over-exposed area. A function that reconstructs these profiles is called *Expand Map*, a smooth function, that determines the level of needed expansion for pixels belonging to areas of high luminance allowing a plausible restoration with available local information. The expand map can be defined as:

$$\Lambda : [0,255]^{w\times h} \rightarrow [0,1]^{w\times h} \tag{5.12}$$

Since Λ is a smooth function, it can reduce contouring during expansion, because expansion follows the low frequency profiles of reconstructed light sources. For example, a smooth gradient is created in an over-exposed region next to a well-exposed one, reducing the step edge that expansion can introduce in this situation.

Note that the expand map leaves under-exposed regions untouched. A reason is that there is no good criteria to restore luminance in very dark regions, such as the creation of light sources in very bright areas. Furthermore, pixels of an image are not well discretised in very dark areas, and these parts have many quantisation artefacts such as blocks and ringing [205] in compressed images.

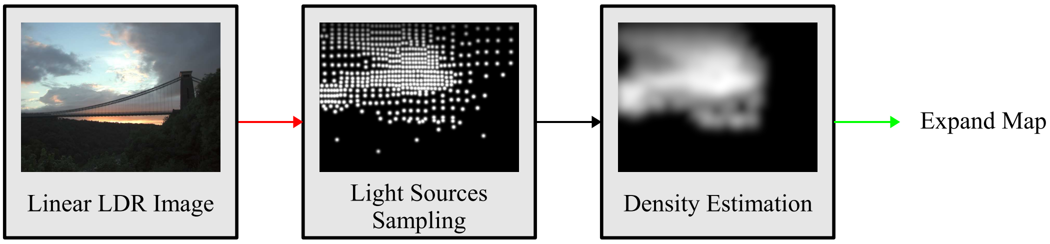


Figure 5.6: *The pipeline for the generation of the expand map for still images. The luminance channel of the LDR image is used to create light sources using an importance sampling algorithm. Then, density estimation is calculated using these samples, generating a smooth field that is the final expand map.*

A simple definition for Λ is to determine areas of high luminance and smooth them with a low pass filter. However, this process can not be based on thresholding, because it is a local function without spatial knowledge which can remove important areas of high luminance, see Figure 5.7 for an example. An approach that preserves all these features is to discretise the image into light source samples, see Section 2.7.2. Then, a smooth field is created using density estimation from these samples. Density estimation is a statistical method that constructs an estimate of a probability density function from observed data [53]. 2D density estimation is calculated for each pixel of Λ using the n-nearest samples inside an influence radius r_s:

$$\Lambda(\mathbf{x}, r_s) = \frac{1}{|\mathscr{P}|V(\mathscr{P})} \sum_{p \in \mathscr{P}(\mathbf{x})} \Psi_p \qquad V(\mathscr{P}) = \pi r_{\max}^2 \tag{5.13}$$

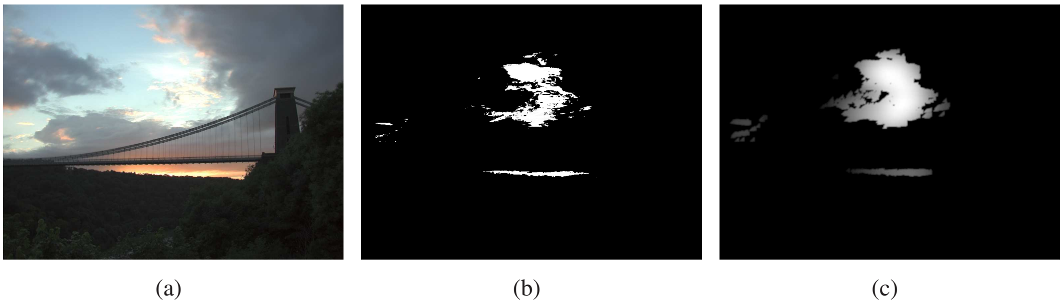
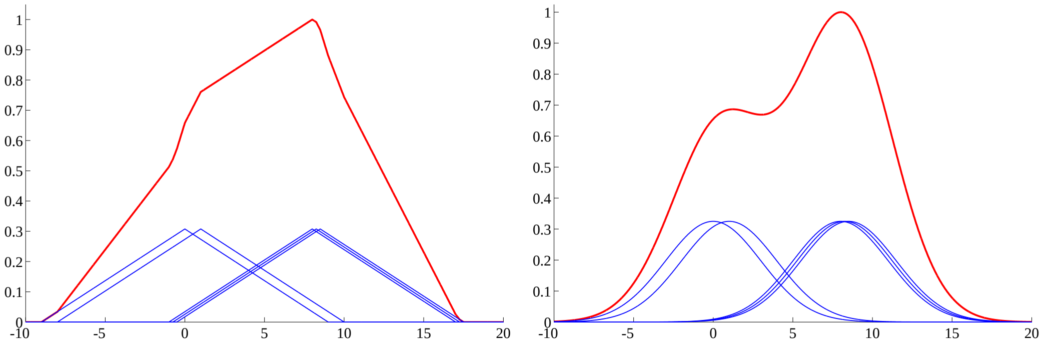

(a) (b) (c)

Figure 5.7: *An expand map generated using thresholding and Gaussian filtering: a) A single exposure LDR image of Bristol Bridge. b) Thresholding on the luminance channel where threshold was set equal to* 0.9 *for determining over-exposed regions. c) A Gaussian filtered image of b) with kernel size of 48. Note that the threshold approach does not include other features of the sky which is an area of high luminance compared to the wood in the bottom. The original HDR image is courtesy of Greg Ward [212]*

where $\mathscr{P} = \{\mathbf{y} : \|\mathbf{x} - \mathbf{y}\| \leq r_s\}$ is the set of samples in the radius r_s, Ψ_p is the luminance of the p-th sample, V is the area of the n-nearest samples that fits a circle, and $r_{\max} \leq r_s$ is the maximum distance from $\mathbf{x}$ to $\mathbf{y}_{\max}$ the farthest sample in $\mathscr{P}$. Parzen windows or smoothing kernels can improve the smoothness of Equation 5.13, obtaining:

$$\Lambda(\mathbf{x}, r_s) = \frac{1}{|\mathscr{P}|V(\mathscr{P})} \sum_{p \in \mathscr{P}} K\left(\frac{\|\mathbf{x} - \mathbf{y}_p\|}{r_{\max}}\right) \Psi_p \tag{5.14}$$



(a)

(b)

Figure 5.8: *An example of density estimation in 1D with different kernels with samples at* $\mathbf{x} = [0.0, 1.0, 8.0, 8.25, 8.50]^\top$*: a) The density estimation using a Cone kernel with* $\alpha = 1/3$*. While* Λ *is plotted in red, partial contributions are plotted in blue. b) The density estimation using a Gaussian kernel. Note that while the global shapes is kept both in a) and b), the shape of the kernel influences local features. For example, the concavity in b) around* $x = 5$ *is not present in a).*

where K is a symmetric smoothing normalised kernel based on distance. This is a decreasing function where $K(x \geq r_s) = 0$. The more common kernels for K are Gaussian and Cone functions. The first is defined as:

$$K_{\mathrm{G}}(x) = \begin{cases} \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2r_s^2}} & \text{if } |x| \leq 1 \\ 0 & \text{otherwise} \end{cases} \tag{5.15}$$

(a)

(b)

(c)

Figure 5.9: *An example of 2D density estimation applied to an LDR version of the Bristol Bridge HDR image: a) The density estimation using a Cone kernel with* $\alpha = 1$*. b) The density estimation using a Gaussian kernel. Note that the estimates in a) and b) look similar, differences are more difficult to find compared to the 1D case. The original HDR image is courtesy of Greg Ward [212].*

Figure 5.8.b and Figure 5.9.c show examples of estimation using K_G. The latter is defined as:

$$K_C(x) = \begin{cases} \frac{1}{\alpha}\left(1 - \alpha|x|\right) & \text{if } |x| \leq 1 \\ 0 & \text{otherwise} \end{cases} \tag{5.16}$$

where $\alpha \in (0, +\infty)$ is a constant that characterises the filter shape, see Figure 5.8.a and Figure 5.9.b for examples of estimation using K_C.

An example application of reconstruction and noise reduction using the outlined expand map is shown in Figure 5.10.

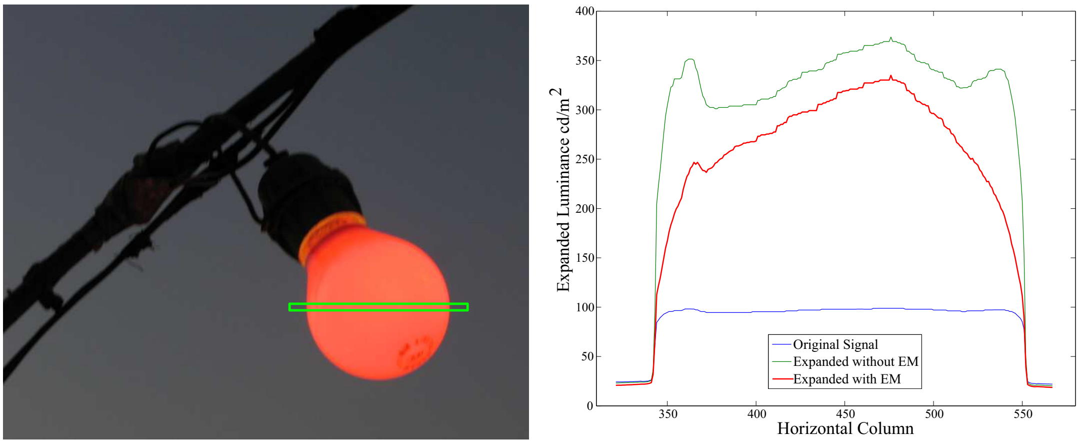
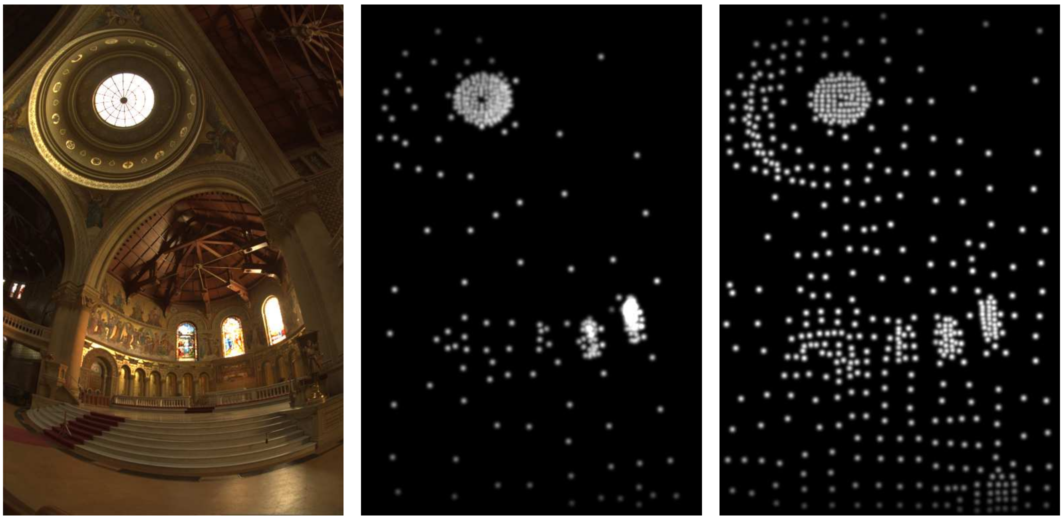


Figure 5.10: *An example of reconstruction and noise reduction using the expand map. On the left side the Red Bulb LDR image, where a row around the light bulb is highlighted in green. On the right side the graph of luminance values of the highlighted row for the original image (blue line), expansion with (red line) and without (green line) expand map. Note that the simple expansion expands noise and does not reconstruct the clipped profile, it keeps the flat clipped region. On the other hand, the expansion with the expand map generates a plausible reconstruction shape profile which reduces expansion of the noise.*

Light Sources Generation

Light sources generation was reviewed in Section 2.7.2. Among the main methods for extracting light source samples, see Section 2.7.2, a suitable one for generating an expand map is MCS [170, 45]. This is because MCS is computationally fast, therefore it is suitable for real-time applications and videos. Moreover, it is straightforward to implement in parallel or on graphics hardware. Other methods such as SIS [4] and k-means sampling (KMS) [97] need iterative relaxations which can be computationally expensive and can not meet real-time deadlines due to their iterative nature. Furthermore, MCS is easier to implement and more computationally efficient than PTS [148], which allows real-time sampling as well.



(a) (b) (c)

Figure 5.11: *An example of generated samples using MCS from Memorial Church HDR image: a) A single exposure of the image. b) Light source samples are calculated from HDR values. c) Light source samples are calculated from LDR values, single exposure in a). Note that the samples distribution shares some features to the one in b). The original HDR image is courtesy of Paul Debevec [44].*

Note that applying a sampling method to an LDR image and to an equivalent HDR one can produce different results. However, some features in the distribution of samples are kept in the LDR image, see Figure 5.11.

MCS is based on median cut colour quantisation by Heckbert [80]. The idea of the algorithm is to divide the image in 2^p regions of equal luminance, where $n = 2^p$ is the number of samples and p is the number of cuts of the image. At the beginning of the algorithm there is only one region, the full image. Then, at each iteration the image is subdivided along the longest dimension such that the luminance is divided evenly in the two sub-images. The process continues until n regions are created. Subsequently, for each region a sample is placed in the centroid of each region with energy equal to the sum of luminance values within the region. An example of the sampling process using MCS is shown in Figure 5.12. Samples are stored in a 2D-tree [177], to improve the efficiency of range queries for $\mathscr{P}$. Note that the whole process can be sped up calculating a summed area table of the image [37].

An heuristic for determining the number of cuts is given by $p = \lceil \log_2 \sqrt{h \times w} \rceil$ where h and w are respectively the height and width of the image. The idea is to generate a sample for

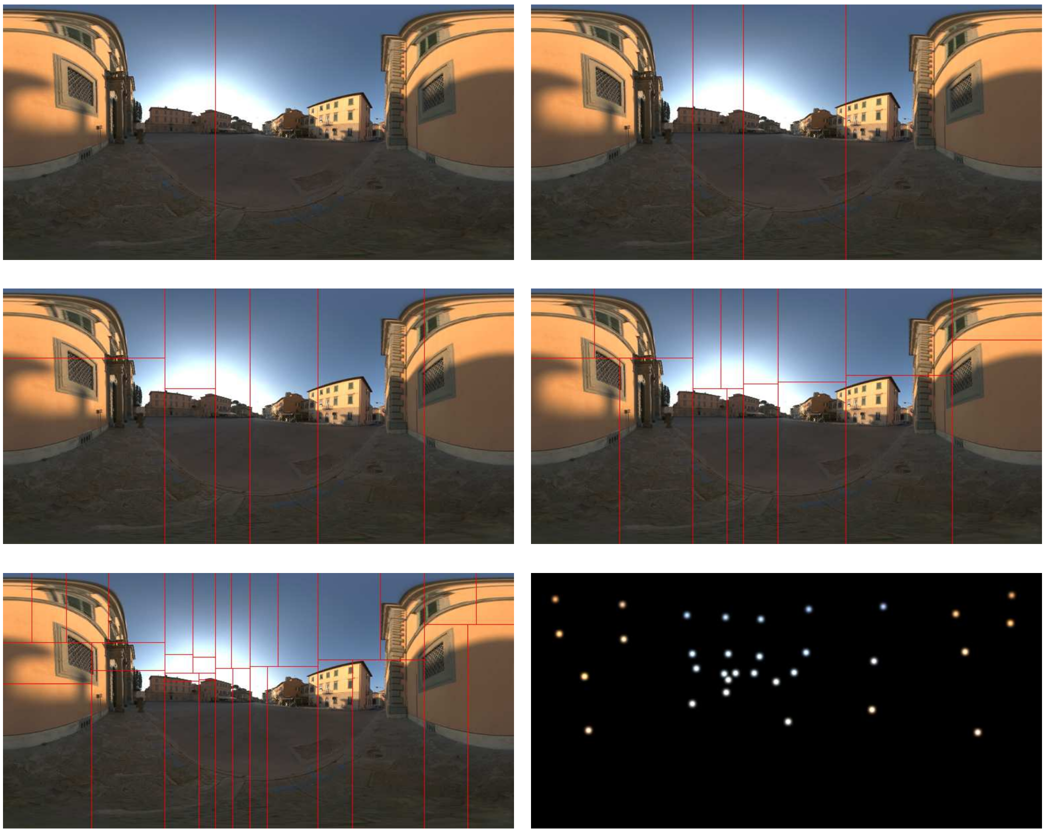


Figure 5.12: *An example of MCS applied to Pisa HDR image for generating 32 lights. From the top to the bottom the increasing number of cuts from 1 to 32, which doubles for each iteration. In the last row on right, there are the final generated samples. The original HDR image is courtesy of Paul Debevec [44].*

each $2^{\frac{p}{2}} \times 2^{\frac{p}{2}}$ region in the case of a uniformly coloured image. This produces enough low frequency information for calculating a smooth Λ. For example, $1,024$ samples are generated for an $1,920 \times 1,080$ image.

A problem with MCS is the generation of few isolated samples. This is due to the presence of large low luminance areas, and few samples are drawn there. However, these can produce isolated areas of high luminance during the evaluation of density estimation, which have to be considered as outliers because they are low luminance ones. A straightforward solution to this problem is to set void density estimation, when few samples are in $\mathscr{P}$, which can be mathematically expressed as:

$$\Lambda'(\mathbf{x}, r_s) = \Lambda(\mathbf{x}, r_s) f(|\mathscr{P}|) \qquad f(x) = \begin{cases} 1 & \text{if } x \geq ns_{\text{Min}} \\ 0 & \text{otherwise} \end{cases} \tag{5.17}$$

where ns_{Min} is the minimum number of needed samples to validate the estimation.

Colours Treatment

Clipped colours in over-exposed areas can be reconstructed employing coloured expand maps. The only modifications is to generate coloured light source samples during MCS, and to extend density estimation to three colour channels, see Figure 5.13 for an example. Note that the risk of colour shifts in well-exposed areas is minimal. This is because light source samples are placed in areas of high luminance where pixels are usually clipped. Therefore, the application of a coloured expand map happens in saturated regions. Furthermore, the colour of the expand map in a region is similar to the one of that region in the LDR image. Therefore, colours are not greatly modified. In the worst case, when a well-exposed region is under the influence of a coloured expand map, there is an increase of colour saturation.

The use of a coloured expand map works well when over-exposed areas are not too large, for example less than 5-10% of the area of the image. Large over-exposed areas tend to produce white samples, so the expand map is grey in those areas.

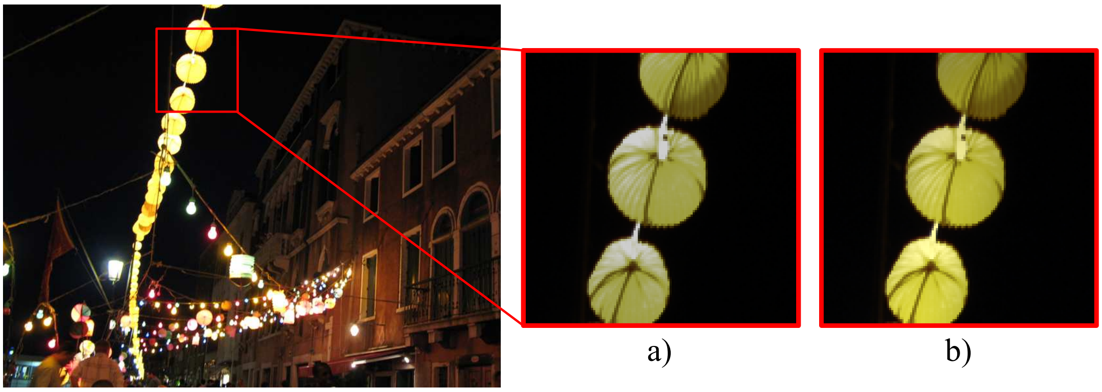


Figure 5.13: *An example of coloured expand map in the Redentore LDR image. On the left side the original image. On the right side two zooms of two differently expanded images around some lanterns in the scene: a) The image was expanded using a luminance expand map. Note that colours inside the lantern are lost. b) The image was expanded using a coloured expand. Note that yellow tint was reconstructed using colours from light source samples.*

5.1.4 Computational Complexity Analysis

An important aspect for an iTMO is its computational complexity for understanding its scalability when applied to images at different resolutions [151]. It is assumed that the number of samples is $\sqrt{n}$ where n is the number of pixels in the image. The computational complexity of the whole framework is calculated as a sum of the contribution of each step, which are:

- **Linearisation**. $\mathcal{O}(n)$: the application of the inverse CRF has a constant computational time cost, $\mathcal{O}(1)$, and it is applied to all n pixels of the image;
- **Range expansion**. $\mathcal{O}(n)$: the application of range expansion function has a constant time cost, $\mathcal{O}(1)$, and it is applied to all n pixels of the image;
- **MCS complexity**. $\mathcal{O}(n \log n)$: in the worst case a median cut moves through all m pixels of the longest regions, which means a $\mathcal{O}(m)$ complexity. Therefore, the number of cuts is $\log\sqrt{n}$ for generating $\sqrt{n}$ samples, assuming the worst case for each cut with a reduction of only one pixel along the longest direction the complexity for MCS is given by:

$$\begin{aligned}\sum_{i=1}^{\log\sqrt{n}}(n-i) &= n\log\sqrt{n}-\sum_{i=1}^{\log\sqrt{n}} i\\ &= n\log\sqrt{n}-\frac{1}{2}\left(\log^2\sqrt{n}+\log\sqrt{n}\right)\\ &= \frac{1}{2}n\log n-\frac{1}{8}\log^2 n-\frac{1}{4}\log n \qquad (5.18)\end{aligned}$$

- **Density Estimation**. $\mathscr{O}(n(k+\log\sqrt{n}))$: a k-nearest neighbours search in a kD-tree costs $\mathscr{O}(k+\log\sqrt{n})$. The evaluation of density estimation costs $\mathscr{O}(k)$. This process is iterated for all n pixels;

- **Linear Interpolation**. $\mathscr{O}(n)$: the application of linear interpolation has a constant computational time cost, $\mathscr{O}(1)$, and it is applied to all n pixels of the image;

Summing all contributions, the final complexity is $\mathscr{O}(n\log(n)+n)$. The main costs in the algorithm are due to the sampling algorithm and the data structure for storing samples, the kD-tree.

5.2 Inverse Tone Mapping for Videos

The method presented in Section 5.1 can produce flickering if it is applied to video sequences for a series of reasons. Firstly, many parameters need to be hand chosen for generating the desired result for each image, which makes the task very tedious for all frames of a video. Secondly, range expansion, median cut, and density estimation are not temporally coherent. Hence, they use only the information from the current image.

This section shows how to extend the algorithm for videos. Moreover, only the parts that need to be modified for temporal coherence are presented. The parts that can be kept from the previous algorithm are the linearisation process and the linear interpolation. The former is a simple application of a function that depends on the device that captured the content, therefore it maintains temporal coherence. The latter is a simple interpolation between the linearised LDR image and the expanded one using an expand map, which final result depends on the temporal coherence of inputs.

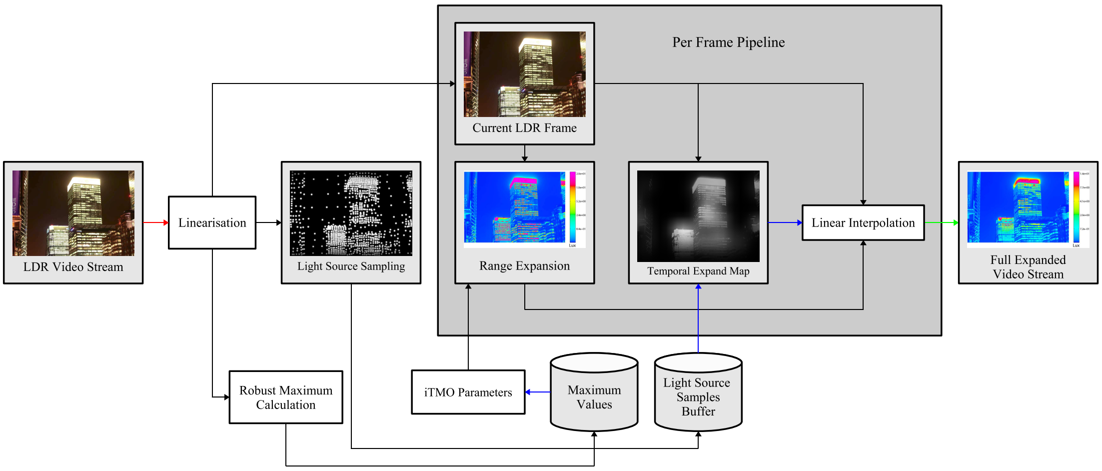


Figure 5.14: *The pipeline of the framework for the iTMO for videos. For each frame of a video, light source samples are generated for the next n frames and stored in a buffer that caches the n backward and forward frames. Then, the current LDR frame is expanded using the expansion function which is the same as in Section 5.1.2, but* $L_{d,\,Max}$ *is smoothed using neighbour's frame* $L_{d,\,Max}$ *to avoid flickering. Subsequently, a temporal coherent expand map is generated using buffered samples and the current LDR frame. Finally, the LDR and the expanded frame are linearly interpolated using the temporal expand map as weight.*

In the case of video sequences the temporal coherence between frames must be taken into account, and the framework has to be accordingly modified, see Figure 5.14. In this new version of the framework, the range expansion has as input smoothed values. This is necessary to reduce flickering during expansion, see Section 5.2.2. Also the density expansion is changed to be temporal coherent. This is achieved by using light source samples from previous and forward frames, see Section 5.2.2.

5.2.1 Temporal Pixel Values Expansion

Assuming that $L_{\text{d, Max}} = 1$, the expansion of pixel values remain the same as in Section 5.1.2. This is because Equation 5.7 does not produce any flickering if β is calculated as in Equation 5.9.

However, flickering can occur if β is computed as in Equation 5.8, because it takes as input the current frame $L_{\text{d, Max}}$, which can vary rapidly from one frame to another. The first step to solve this problem is to calculate robustly $L_{\text{d, Max}}$. This is achieved by utilising percentiles, for example the 95-th percentile gives a robust estimate of the maximum value. This avoids problems introduced by noisy pixels. Then, the second step is to filter $L_{\text{d, Max}}$ to have smooth values over frames, avoiding flickering:

$$L'_{\text{d, Max}} = L_{\text{d, Max}} \otimes G_s \tag{5.19}$$

where $L'_{\text{d, Max}}$ is the filtered LDR maximum, and G_s is a 1D Gaussian kernel with variance σ_s. From preliminary tests, $\sigma_s = 1$ results to be a suitable value, see Figure 5.15.

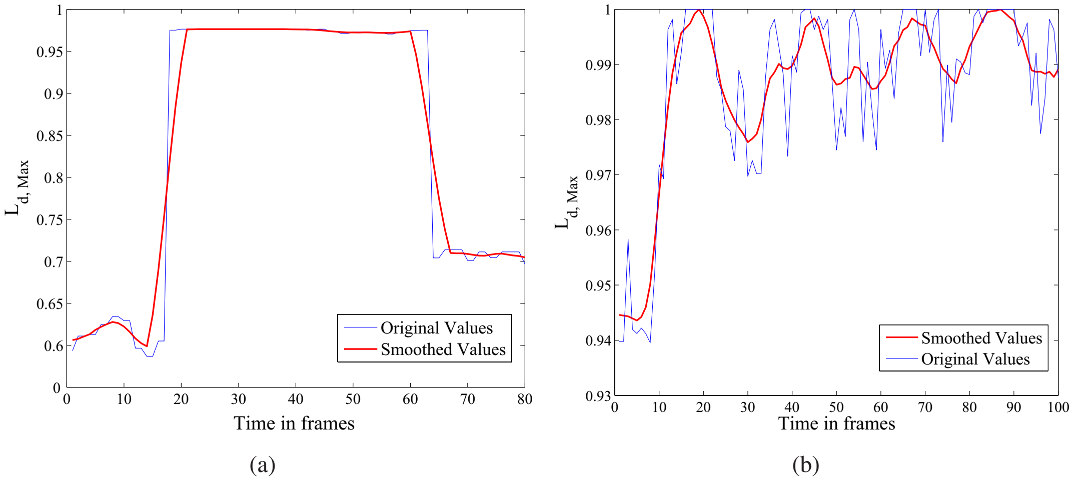


Figure 5.15: *An example of filtering $L_{d,\ Max}$ for two scenes. While the blue line is the original signal, the red one is for the filtered one: a) An HD BBC sequence [19]. b) A flame sequence from Fedkiw's website [62]. Note that large fluctuation that can cause flickering are smoothed.*

5.2.2 Temporal Expand Map

A naïve application of the method in Section 5.1.3 to a video sequence can produce flickering in the expand map, and therefore flickering in the final expanded video. This is because light source samples are generated independently for each frame, and a sample can pop in and out in small regions for two next frames. Furthermore, density estimation is performed with only samples of the current frame.

In order to solve this problem of incoherent samples median cut can be extended to the temporal domain, see Figure 5.16 for the pipeline. This would allow to keep 2D density estimation, which would use samples of the current frame which are temporally created. However, due to its nature, MCS needs to have the uncompressed video or its summed-area table [37] in the main memory. This is quite impractical due to hardware constraints. For instance, an 8 seconds video sequence in high definition resolution, $1{,}920 \times 1{,}080$, at 24 frames per second would require more than 1GB of free memory. Clearly, this solution is not very space efficient.

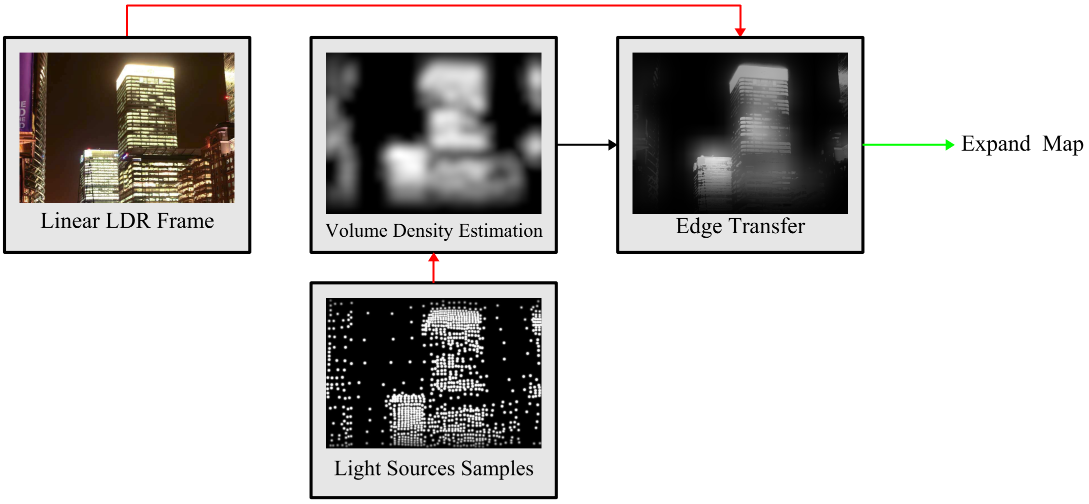


Figure 5.16: *The pipeline for the generation of temporal expand maps. The volume density estimation is calculated using light source samples from a buffer that caches them from the n-th backward and forward frames from the current one. Then, edges from the current frame are transferred to the result of volume density estimation obtaining the final temporal expand map. The edge transfer alleviates ghosting effects and enhances edges.*

An alternative more efficient memory-wise approach is to apply MCS separately to each frame, but to extend density estimation to temporal domain. Firstly, samples for all frames are stored onto a 3D-tree, where the third coordinate is set to the frame number or time. Secondly, temporal density estimation is evaluated gathering samples from other frames using a volume shape, in a similar approach to volume photon mapping [90]. To avoid incoherent samples from frames that change completely from the previous one, videos are cut in shots. A shot is a series of frames where changes between frames are smooth. These can be determined using an histogram video partition technique [74].

The temporal density estimation ensures that flickering is alleviated if not removed. Note that a small radius in the temporal domain is needed, because objects can rapidly move in a video, invalidating samples. Therefore, an ellipsoidal volume was chosen rather than a spherical one, otherwise features are lost in the motion of objects, see Figure 5.17 for a 2D temporal example. The density estimation is filtered using a cone kernel for removing unwanted noise. The final equation for density estimation is given by:

$$\Lambda(\mathbf{x},t,r_s,r_t) = \frac{1}{|\mathscr{P}|V(\mathscr{P})} \sum_{p\in\mathscr{P}} K_C\left(\frac{\|\mathbf{x}-\mathbf{y}_p\|}{r_{\max}}\right)\Psi_p \qquad V(\mathscr{P}) = \frac{4}{3}\pi r_s^2 r_t \tag{5.20}$$

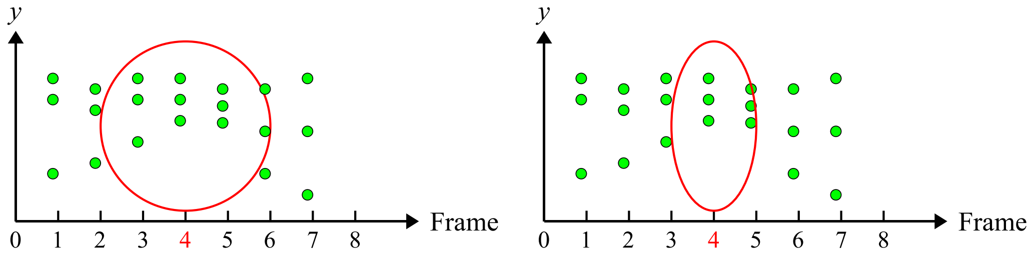


Figure 5.17: *An example of temporal density estimation applied to a 1D video sequence. The density estimation is calculated at frame 4 using a temporal radius of 1 frames in backward and forward. Two different shapes, a circle and an ellipse, are used in the density estimation. On the left side a circle has as radius the maximum between temporal and spatial radius, otherwise the spatial radius is too small. Therefore, a circle includes too many samples. This produces a too smoothed motion losing features. On the right side an ellipse has a shape that avoids to gather too many samples with a large spatial radius. Therefore, original features of the motion are kept better than using a circle.*

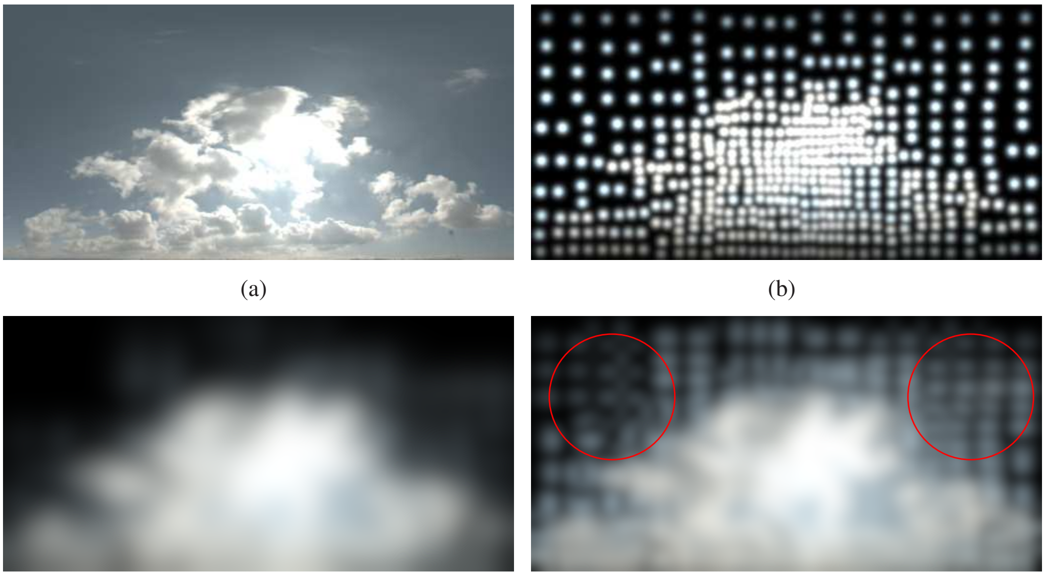
where $\mathscr{P} = \left\{ \mathbf{y} : (\mathbf{x}-\mathbf{y}) \cdot [r_s, r_s, r_t]^{-\top} \leq 1 \right\}$ is the set of points in the ellipsoid, and r_s and r_t are respectively the semi-axis in space and time. V is the volume of the ellipsoid that fits the n-nearest samples, and $r_{\max} \leq \max(r_s, r_t)$ is the maximum distance from $\mathbf{x}$ to $\mathbf{y}_{\max}$, the farthest sample in $\mathscr{P}$. As in Section 5.1.3 the estimate is clamped if there are more than ns_{Min} samples.

Automatic Estimation of Density Estimation Parameters

The main problem for video sequences is how to estimate the parameters for density estimation which are: r_t, r_s, ns_{Min}. Note that an optimal estimation for the first few frames does not assure that they will be kept during the video sequence. Therefore, some heuristics to avoid tedious hand selections and possible flickering are proposed.

The first parameter, r_t, can be calculated in a progressive way. For example, the ellipsoid is expanded in the temporal direction until variations in the number of samples is very large. This means that samples are completely changed, therefore they are no longer valid. However, it can be found empirically, analysing various videos, that $r_t \in [3, 5]$ is large enough, avoiding further computations for expansion of ellipsoid along the time direction.

The second parameter, r_s, affects more the estimation, because it determines the number of samples for the current frame. These samples represent a large percentage in the total estimation. As for the case of r_t, r_s should be expanded until enough samples are reached. On the other hand, this expansion has to be limited to an upper bound. This can lead to an indefinite



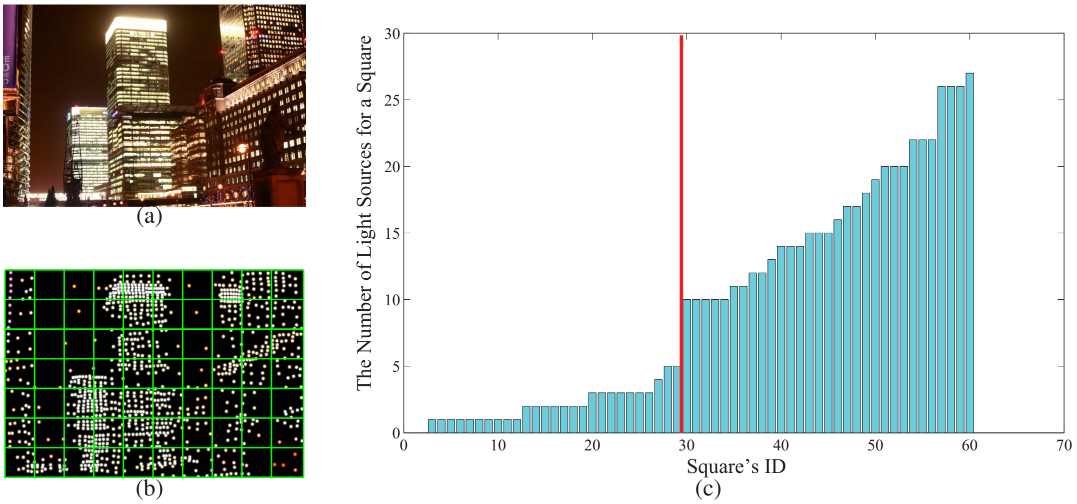
(a) (b)

(c) (d)

Figure 5.18: *A comparison between automatic and non automatic density estimation for the 20-th frame of the sky sequence: a) The LDR frame. b) The light sources generated by median cut. c) The result of the density estimation using automatic parameters estimation. d) The result of the density estimation using parameters which were estimated by an user for the first frame. Note inside red circles the expand map is estimating wrongly the density of light sources generating areas to expand. The original HDR image is courtesy of Paul Debevec [187].*

expansion in areas without samples, which will produce a wrong estimation. The upper bound $r_{s,\max}$ is calculated from ns_{Min} the minimum number of samples needed to terminate the expansion of r_s. ns_{Min} is calculated using a simple heuristic. For the current frame, the XY plane is divided in a regular grid of cells with length $l_{square} = \max(w,h)/k$, where $k = \log_2 n_{samples}$, w and h respectively are the width and height of the frame. Subsequently, the number of samples for each region is counted and sorted in an array, H. At this point, ns_{Min} is calculated as the value in H with the biggest slope between its predecessor and its successor, see Figure 5.19. The idea is that the presence of a large slope indicates the difference from cells with few or no samples to cells with many samples. However, it may happen that there is a constant slope, in this case this metric will not work. To solve this problem ns_{Min} is set equal to the median value in H. Finally, $r_{\text{s,max}}$ is calculated as the distance from the centre of the median square to the farthest sample in it.

Note that these heuristics can generate values that vary at each frame potentially causing flickering. To avoid it, these values are filtered similarly to $L_{\text{d, Max}}$ in Section 5.2.1 using a 1D Gaussian filter.



(a) (b) (c)

Figure 5.19: *An example of automatic parameter estimation: a) The image used for the estimation. b) Generated samples from a) which are divided by a grid. c) The graph of number of samples for each cell of the grid, with division line for ns_{Min} (in red).*

Figure 5.18 shows the advantages of automatic parameter estimation using the sky sequence. As can be seen, artefacts appear in the expanded HDR frame where parameters were not chosen correctly.

Edge Enhancement

Once the volume density estimation is calculated, it can be used for interpolation of the LDR and the expanded frame. However, volume density estimation can introduce some ghosting effect even if the temporal axis r_t is small including only a few frames. Moreover, a too smooth field can reduce contrast around edges in the form of glare. A simple solution to these problems would be to employ an edge stop function as in Rempel et al.'s work [171], but this function needs a threshold for defining edge and it can potentially produce flickering.

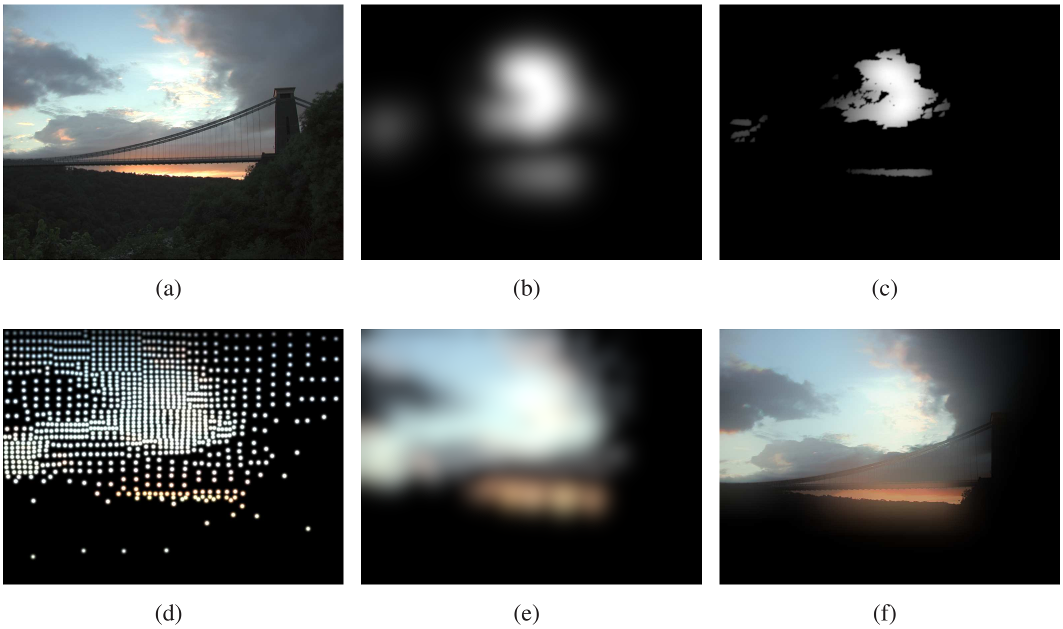
Another straightforward solution is to transfer edges, which are present in the original frame, to the expand map. The idea is that keeping original edges reduces the ghosting effects which can be more noticeable at the boundaries of a moving object. Furthermore, edges are enhanced. The transferring of edges is achieved by employing the cross/joint bilateral filtering, see Appendix A for more details. The filter is defined as:

$$\Lambda_E(\mathbf{x}) = \frac{1}{k(\mathbf{x})} \sum_{\mathbf{y} \in \mathrm{W}} \Lambda(\mathbf{x}) G_S(\|\mathbf{y}-\mathbf{x}\|) G_R(\|L_\mathrm{d}(\mathbf{x}) - L_\mathrm{d}(\mathbf{y})\|)$$

$$k(\mathbf{x}) = \sum_{\mathbf{y} \in \mathrm{W}} G_S(\|\mathbf{y}-\mathbf{x}\|) G_R(\|L_\mathrm{d}(\mathbf{x}) - L_\mathrm{d}(\mathbf{y})\|)\|) \tag{5.21}$$

where $\Lambda(\mathbf{x})$ is the final expand map, W is the filtering window, G_R and G_S are two Gaussian functions, L_d is the luminance of the LDR frame used as a guidance map, in this case the luminance of the image.

Figure 5.20 compares the proposed expand map using joint bilateral filtering Λ_E, and the one produced with Rempel et al.'s algorithm [171].


(a) (b) (c)

(d) (e) (f)

Figure 5.20: *A comparison between methods for generating the expand map. In the top row the method presented in [171]: a) A single exposure LDR version of Bristol Bridge HDR image. b) A Gaussian filtered luminance channel for a) after thresholding. c) The final expand map with the application of an edge stopping function to b). In the bottom row our method: d) Light source samples for a) generated using median cut algorithm. e) The expand map, for each colour channel, which is the result of the density estimation applied to d). f) Final expand map for a) applying cross/joint bilateral filter to e) using luminance of a) as guidance. The original HDR image is courtesy of Greg Ward [212].*

The calculation of the temporal density estimation is the main cost during the calculation of the expand map. To speed-up computations, MCS and density estimation can be generated using a down-sampled version of the luminance channel, such as a quarter of the area. The up-sampling to full frame size can be applied during the edge transfer using the joint bilateral up-sampling [98]. In this way, up-sampling and joint bilateral filtering are performed at the same cost in one filtering operation. Therefore, Equation 5.21 can be modified as:

$$\begin{aligned}\Lambda_E(\mathbf{x}) &= \frac{1}{k(\mathbf{x})}\sum_{\mathbf{y}\in \mathrm{W}} \Lambda(\mathbf{x}\downarrow) G_S(\|\mathbf{y}\downarrow - \mathbf{x}\downarrow\|) G_R(\|L_{\mathrm{d}}(\mathbf{x}) - L_{\mathrm{d}}(\mathbf{y})\|)\\ k(\mathbf{x}) &= \sum_{\mathbf{y}\in \mathrm{W}} G_S(\|\mathbf{y}\downarrow - \mathbf{x}\downarrow\|) G_R(\|L_{\mathrm{d}}(\mathbf{x}) - L_{\mathrm{d}}(\mathbf{y})\|)\|) \end{aligned} \tag{5.22}$$

where $\downarrow$ is an operator which maps L_{d}'s coordinates to Λ's coordinates. While the variance value for G_S depends on the down-sampling factor, see Paris et al. [153], the one for G_R depends on which edges are desired to transfer. In the implementation of the framework weak edges are not kept, to avoid to transfer noise, therefore $\sigma_R = 0.1$, see Paris et al. [153].

This operation does not degrade the quality of the expand map because it is a smooth field, and up-sampling operations usually deteriorate more edges than low frequencies, see Figure 5.21 for an example.

(a) (b)

Figure 5.21: *An example of expand map calculated at low resolution: a) An expand map calculated at full resolution. b) An expand map calculated at one 16th of the original resolution. Note that the differences are difficult to notice due to the smooth nature of* Λ.

5.2.3 Computational Complexity Analysis

The computational complexity of the whole framework for videos is calculated as a sum of the contribution of each step for t frames. The complexity for a single frame is presented, and only for the steps that changed compared to the still images version:

- **Automatic Parameter Estimation**. $\mathcal{O}(\sqrt{n} + \log^2 n \log\log^2 n)$: sorting the cells of the grid has a computational cost $\mathcal{O}(\log^2 n \log\log^2 n)$. Where $m = \log^2 \sqrt{n}$ is the number of cells, and state of art sorting algorithm has computational complexity $\mathcal{O}(m \log m)$ [172]. The maximum slope or median derivate cost $\mathcal{O}(m)$. Finally, r_s has computational complexity $\mathcal{O}(\sqrt{n})$ in the worst case, when all samples are in the same cell.

- **Density Estimation**. $\mathcal{O}(n(k + \log p))$: the computational complexity is similar to the one for still images. The total number of samples is $p = \sqrt{nt}$ instead of $\sqrt{n}$. Hence, the 3D-tree is built using samples from all frames.

- **Joint Bilateral Up-sampling**. $\mathcal{O}(nl^2)$: complexity depends on the size of the kernel $l = \lceil 5 \max(\sigma_s, \sigma_r) \rceil$. Note that the filter can be approximated to speed computations up. For example Pham and van Vliet's method [161] applies two 1D separate bilateral filters achieving a $\mathcal{O}(nl)$ complexity, but quality decreases for large kernels. This problem can be solved using Paris and Durand's approximation [152], which employs signal processing techniques and has $\mathcal{O}\left(\frac{|S|}{\sigma_s^2}\frac{|R|}{\sigma_r^2}\log\left(\frac{|S|}{\sigma_s^2}\frac{|R|}{\sigma_r^2}\right)\right)$ complexity, where S and R are respectively the spatial and range domain.

Summing all contributions, including the unchanged ones from Section 5.1.4, the final complexity for a video is $\mathcal{O}(tnl^2 + tn\log n + tn\log(k + \sqrt{nt}))$. The main costs are still given by MCS and the data structure for samples, the 3D-tree. However, joint bilateral up-sampling can have a substantial weight, especially if it is not sped up.

5.3 Real-Time Inverse Tone Mapping using Graphics Hardware and Multi-Core CPUs

An efficient implementation of algorithms presented in Section 5.1 and Section 5.2 can based on multi-core CPUs and graphics processing units (GPUs) to speed-up calculation, and make it possible to generate real-time expansion for images and videos, see Appendix C for an introduction on GPUs.

The idea is to split computations between a multi-core CPU and a GPU. This is achieved by leaving pure image processing computations to the GPU such as linear interpolation, range

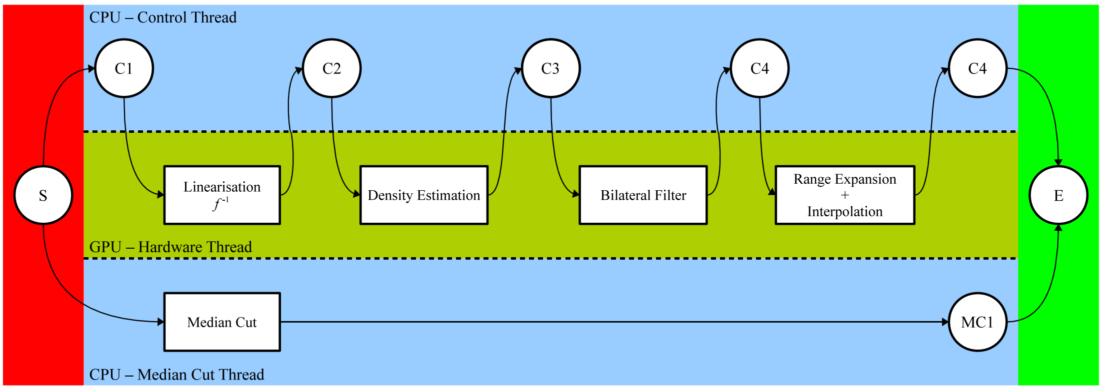


Figure 5.22: *The computation flow showing the cooperation between CPU and GPU threads for the iTMO. The first thread in the top*

expansion, linearisation, and density estimation. Compuations which are difficult to map on graphics hardware are left to the CPU, such as automatic parameter estimation and MCS which requires a stack to save the cuts and arbitrary writing access to the memory (still not efficient on modern GPUs). GPU operations can be classified in two kinds:

- **Simple Operation**: needs only to apply a function f to all pixels of the image, without further processing. Linearisation, interpolation, and range expansion are simple operations.
- **Multi-pass Operation**: needs more passes to be completed, these can be simple operation or a pass which renders primitives. The generation of an expand map needs more passes such as the evaluation of density estimation and the joint bilateral up-sampling.

5.3.1 Simple Operations

Linearisation was implemented copying the kernel of the software implementation in a pixel shader, see Listing 1 for the shader source code. Range expansion and interpolation were combined together in a single operation which saves a rendering pass, see Listing 2 for the shader source code.

5.3.2 Point-Based Density Estimation

Evaluation of density estimation on the CPU is computed in two steps. Firstly, samples around the evaluation point are collected using a spatial query. Secondly, Equation 5.14 is evaluated using gathered samples. However, a spatial query on a GPU can be not very efficient to implement, because hierarchical data structures can not be naturally mapped on current graphics hardware.

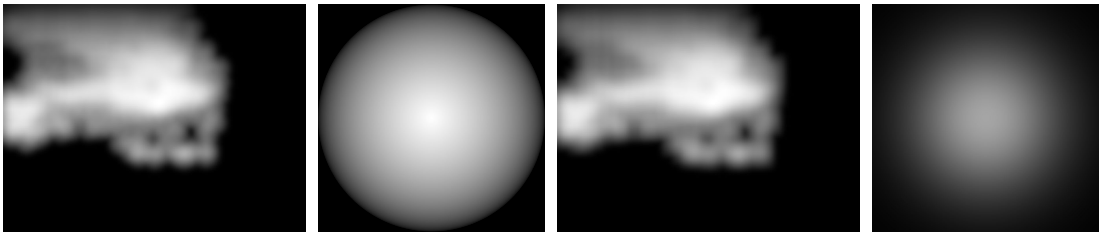


Figure 5.23: *An example of density estimation on GPU using Cone and Gaussian kernel applied to Bristol Bridge LDR image: a) Density estimation using a Cone kernel. b) The Cone kernel with* $\alpha = 1$ *used in a) in the domain* $[-1,1] \times [-1,1]$. *c) Density estimation using a Gaussian kernel. d) The Gaussian kernel with* $\sigma = 1$ *used in c) in the domain* $[-3,3] \times [-3,3]$.

GPUs are very fast in drawing primitives (fill-rate is 38 Gpixels per second for G80 class boards [145]), especially for simple primitives such as points. Furthermore, they have built-in instructions for texture and filtering operations. Therefore, the evaluation of density estimation can be redesigned to exploit these capabilities: fast primitive rendering, and texture support. The idea is to render a textured point for each light source sample, where the size of the point is equal to the radius r_s. This covers all pixels under the influences of that sample. The texture used for these points is a discretised smoothing kernel used in Equation 5.14, which allows to perform filtering, see Figure 5.23. Furthermore, the accumulation of values, when two points are overlapped, is achieved by disabling the Z-buffer and enabling the Alpha blending [6].

Moreover, the implementation on GPU is straightforward, because there are only two steps to compute: load samples into a vertex buffer and render points from the vertex buffer using a short shader, see Listing 3 for the shader source code.

The method described before is designed for 2D still images, but the extension to videos is straightforward and it requires only a few modifications. The first step is to add samples from backward and forward frames to the vertex buffer used for rendering. Furthermore, each sample

```
sampler2D LDR_Image;
float inverse_gamma;
float4 poly1[3], poly2[3];

float4 PixelShaderCRF(float2 texCoords)
{
        float4 LDR=tex2D(LDR_Image, texCoord);
        float4 ret=float4(0.0f,0.0f,0.0f,1.0f);

        for(int i=0;i <3; i++)
        {
                //Inverse CRF using a quintic
                ret[i]=((poly1[i].x*LDR[i]+poly1[i].y)*LDR[i]+poly1[i].z)*
                         LDR[i]+poly1[i].w;
                ret[i]=(ret.x*LDR[i]+poly2[i].x)*LDR[i]+poly2[i].y;
        }
        return ret;
};

float4 PixelShaderGamma(float2 texCoords)
{
    return pow(tex2D(LDR_Image,texCoords),inverse_gamma);
};
```

Listing 1: *The pixel shaders for linearisation of the LDR image. PixelShaderCRF linearises an LDR image using a CRF fitted into a quintic polynomial which coefficients are stored in* `poly1` *and* `poly2`. *This method can be used for known cameras, when the CRF can be measured, or for unknown ones using techniques in Section 3.1.1. PixelShaderGamma linearises an LDR image applying the inverse gamma function, which can be employed for DVDs and television programs [89].*

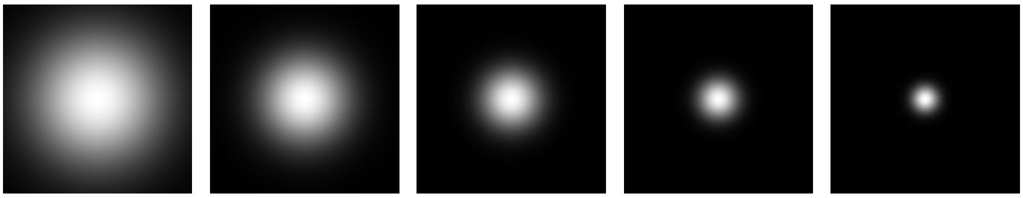


Figure 5.24: *An example of flattened 3D texture for storing a Gaussian temporal kernel using* $r_t = 5$. *From left to right slices for the current frame to the last forward/backward frame. Note that the kernel from backward and forward frames are symmetric, so there is the need to only store only one direction.*

```
1  sampler2D LDR_Image;
2  sampler2D ExpandMap;
3
4  float  Lwhite2, Lwhite2beta;
5  float2 SatValues;
6  float3 LUMINANCE={0.2126,0.7152,0.0722};
7
8  float4 PS_Renge_Interpolation(float2 texCoord): COLOR
9  {
10         //Fetch an LDR pixel
11         float4 LDR=tex2D(LDR_Image, texCoord);
12         float Ld=dot(LDRI.xyz, LUMINANCE);
13         //Expansion
14         float tmp=Ld-1;
15         float Lw=Lwhite2beta*(tmp+sqrt(tmp*tmp+4*Ld/Lwhite2));
16         //Expand Map
17         float4 emap=tex2D(ExpandMap, texCoord);
18         //Linear Interpolation
19         float Lfinal=Lw*emap.x+Ld*(1-emap.x);
20         //Saturation
21         tmp=Ld/Lfinal;
22         tmp=tmp*(3*tmp-2*tmp*tmp);
23         float sat=SatValues.x*(1-tmp)+SatValues.y*tmp;
24         return pow((LDR/Ld),sat)*Lfinal;
25 };
```

Listing 2: *The pixel shader for the range expansion and linear interpolation. The first step is to load from memory the current pixel of the LDR image (line 11), and calculate the luminance (line 12). Then, range expansion is performed using Equation 5.7 (line 14-15), for speeding calculations up some constants are used. For example,* `Lwhite2beta`=$L^2_{white}\beta$, *and* `Lwhite2`= L^2_{white}. *Afterwards, linear interpolation is performed (line 19) using the expand map previously loaded from memory (line 17). Finally, saturation is performed (line 21-24) using Equation 5.10 and Equation 5.11, where* `SatValues.x`= S_{Max} *and* `SatValues.y`= S_{Min}.

```
1  sampler3D texFilter;
2
3  float PixelShaderPointSplatting(float mc_sample_val, float3 texCoords)
4  {
5      //Fetch to precomputed filter
6      float4 val=tex3D(texFilter,texCoords);
7      //Filter value times sample value
8      return val.x*mc_sample_val;
9  };
```

Listing 3: *The pixel shader for 2D/3D density estimation. The shader simply draws all pixels of each textured point. The first step is to load from memory the texture used for encoding the kernel (line 6). Then, this value is multiplied times the luminance power of the sample (line 8) which is stored in the vertex buffer. Note that* `texCoords.z` *refers to the frame in time, for 2D evaluation* `texCoords.z`= 0.

in the vertex buffer has an attribute $t \in [-5, 5]$, which identifies the frame of the sample, where $t = 0$ is the current frame. The second step is to discretise the 3D smoothing kernels. The idea is to store each slice in time of the 3D kernel into a 3D texture, see Figure 5.24 for an example. When textured points are rendered, the access to the correct time slice of the 3D texture is achieved using the extra attribute t added to each sample, see Listing 3 for the shader source code.

Note that the CPU version needs all frames to create the 3D-tree used for the spatial query. The GPU implementation needs only few backward and forwards frames, up to 5, which makes it suitable for real-time streaming of content.

Separate Joint Bilateral Up-Sampling

Joint bilateral up-sampling is a non-linear filter similar to a bilateral filter, see Appendix A, and it can not be computed separately as in the case of a Gaussian filter. Pham and van Vliet [161] proposed an approximation of the filter that produces good and indistinguishable results when $\max(\sigma_R, \sigma_S)$ is not too large. For example a maximum variance up to $8 - 10$ can produce acceptable results [161, 30].

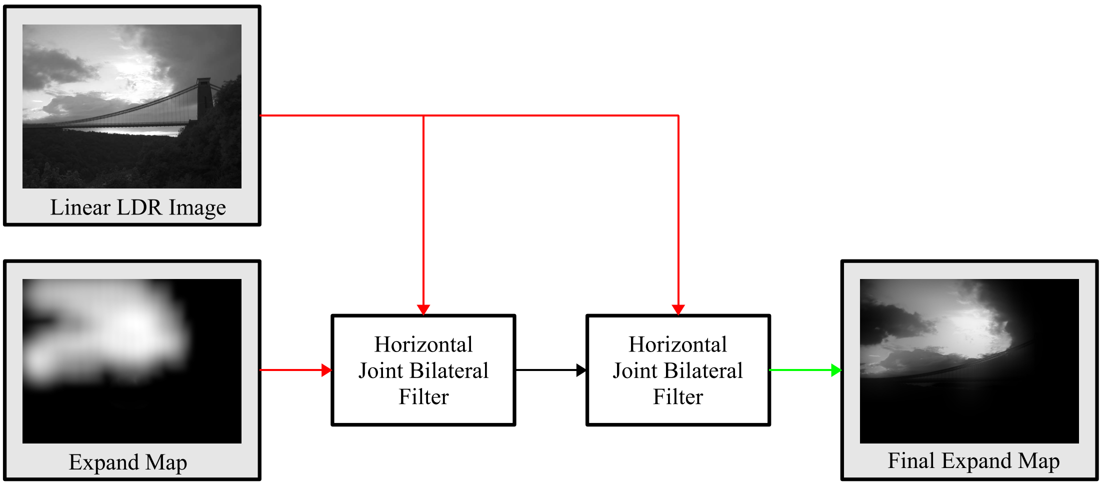


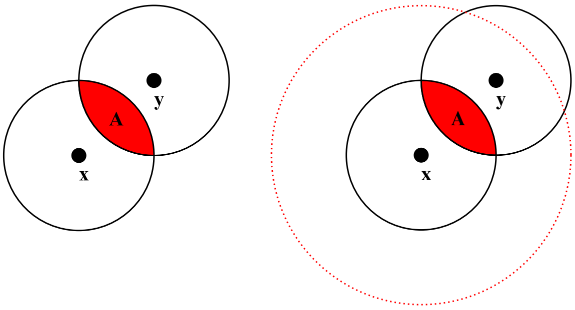
Figure 5.25: *The scheme of Joint Bilatereal Up-Sampling on the GPU. Two separate 1D joint bilateral filters are used to transfer edges from the LDR image to the expand map. The first filter is applied to columns and the second one to rows.*

To speed-up computations the GPU version of the expansion algorithm uses a separate joint bilateral up-sampling inspired by the separation technique of Pham and van Vliet, see Figure

5.25 for the scheme of the up-sampling method. The approximation applies two 1D joint bilateral up-samplings, the first one is applied horizontally and the second one vertically on the result of the first one.

Light Sources Samples Clamping

On the GPU, it is difficult to create a clamping such as on the CPU version presented in Equation 5.17. In fact, on the CPU this operation can be done immediately after the spatial radius query, because it returns the number of samples in the volume, and this avoids performing a density estimation evaluation. However, using the point-based approach the density estimation is performed as a first step, and it is not possible to know in advance the number of samples that will contribute to estimate a pixel. Therefore, there is the need to count these using a second channel in the kernel (equal to 1) during the density estimation, and to store them. Furthermore, a second pass is needed to perform the clamping. As can be noticed this process would slow down performances, because an extra channel in the expand map and in the precomputed kernel, and a second pass are needed to be added. To have a more efficient solution in terms of computational time and memory, an approximated solution is to perform clamping directly on samples before rendering textured points, this operation is called precomputed clamping. For each sample $\mathbf{x}$, $\mathscr{P} = \left\{\mathbf{y} : \|\mathbf{x}-\mathbf{y}\| < r_s\right\}$ is computed. Then it is tested if the number of samples in $\mathscr{P}$ is enough for a density estimation in its area, $|\mathscr{P}| \geq ns_{\text{Min}}$.



(a) (b)

Figure 5.26: *An example of precomputed clamping: a) The precomputed clamping using r_s removes the sample* **y** *from $\mathscr{P}$ for area* **A** *in red. However,* **A** *is influenced for splatting by* **y** *. b) The use of $2r_s$ allows to include the sample* **y** *in $\mathscr{P}$ for the precomputed clamping.*

However, $\mathscr{P}$ has to be calculated using $2r_s$ instead of r_s, because it is possible to exclude some samples for some areas, see Figure 5.26.a and Figure 5.26.b.

5.3.3 Computational Complexity and Timing

The computational complexity of the whole framework the for GPU implementation is calculated as sum of the contribution of each step for t frames. Similarly to Section 5.2.3, only the complexity for a single frame will be presented, and for the steps that changed due to the implementation on the GPU:

- **Density Estimation**. $\mathcal{O}(n_k p)$: the density estimation is achieved using textured points. A textured point needs to be drawn for n_k pixels, the resolution of the kernel texture. This has to be done $p = \sqrt{n}$ times, the number of samples, assuming the worst case where clamping did not remove samples. In the case of the temporal density estimation, if only samples from 3-5 frames backward and forward are used, the complexity does not change, because they can be considered constant and they are absorbed in the notation.
- **Joint Bilateral Up-sampling**. $\mathcal{O}(nl)$: the separate joint bilateral up-sampling costs $\mathcal{O}(nl)$ where $l = \lceil 5\max(\sigma_s, \sigma_r) \rceil$.

Computational complexity on the GPU can be summarised in two parts, one for still images and another for videos. Summing up common and unchanged contributions from Section 5.1.4 and Section 5.2.3, the results are:

- **Still Images**: $\mathcal{O}(n + n\log n + n_k\sqrt{n})$.
- **Videos**: $\mathcal{O}(t(n\log n + nl + n_k\sqrt{n}))$.

In the case of the GPU implementation, timed real-time performances are more important than computational complexity. Therefore, the CPU and GPU version for still images and video were evaluated. While the CPU version was implemented using Matlab R2007b with enabled multi-threading, the GPU one was implemented in C++ using Direct3D9c for accessing to GPU capabilities. Both versions were timed on an Intel Extreme Quad core at 2.4 Ghz, with 2Gb of main memory, and a GeForce 8800GTX with 768 Mb of memory under Windows XP-32 SP3.

The CPU and GPU versions were timed for still images and videos, the results are shown in Table 5.1. As can be seen the GPU version can achieve on average 28.44 fps at high definition (HD) resolution ($1,920 \times 1,080$) while the CPU version can achieve 1 fps at low resolution (360×240). The main bottleneck of this implementation is the low band for transferring frames

	Static Images		Videos	
Image Resolution	CPU (sec)	GPU (fps)	CPU (sec)	GPU Full-Screen (fps)
360×240	0.51	95.32	0.99	83.89
720×480	1.85	69.91	3.56	60.97
1280×720	7.73	53.82	11.68	47.39
1920×1080	26.93	30.59	29.27	28.44

Table 5.1: *The results of the performances of the CPU and GPU algorithm for static images, and video for High Definition televisions resolutions. The performances for the video extension is meant for a frame using as test a video sequence of 60 frames taken from BBC HD Gallery [19].*

from the main memory to the GPU's one, which is more an issue when large images such as at HD resolution, because less memory is available for buffering the video stream. However, the algorithm can expand content in real-time for DVDs and HD content which are at the time of writing the standard for entertainment. Note that the difference between the GPU version for still images and videos is quite small, on average 7-8fps. This is due to the fact that the estimation of parameters is not an computationally expensive task. Moreover, the use of more samples in the density estimation (on average 8 times more for frame in the sequence in Table 5.1) causes a small slow-down, because rendering textured points is a built-in computationally efficient function of modern GPUs.

5.4 Summary

In this chapter, three algorithms for expanding LDR content to HDR content were presented. The first algorithm is designed for static images and needs user intervention for the selection of the parameters. The second one extends the first to video sequences, proposing heuristics to estimate parameters for the model and to avoid temporal flickering. Finally, the third algorithm is a GPU implementation of the other two, where some approximations are introduced for porting the algorithm onto graphics hardware. In conclusion, the GPU algorithm approximates with both previous algorithms, considerably speeding-up computations and achieving real-time frame rates.

Chapter 6

Validation of Inverse Tone Mapping Operators

Recently, the development of inverse tone mapping produced several iTMOs with different features, see Chapter 3. Therefore, there is a need to determine the quality performances of these to understand which iTMO is more suitable than another for a given image and task. Moreover, the analysis of performances can help to highlight important features.

In this chapter a validation of the main iTMOs is presented for two main applications: visualisation of content and IBL. The chapter is divided into two parts. The first one presents a psychophysical study based on the pairwise comparisons [15]. The second part describes a study based on computational metrics [16, 18, 17].

6.1 Inverse Tone Mapping Operators

For both the first and second part of the chapter, five iTMOs were selected which vary in complexity and computational cost. One of the main goals was to determine whether more complex models are required to solve the LDR to HDR problem. The five operators used in the comparisons are:

- Akyüz et al.'s operator [7] (**A**): is a global linear expansion.

- Banterle et al.'s operator Section 5.1 (**B**): is a non linear inverse TMO, that expands the range. To improve reconstruction and reduce noise, an expand map is used to expand only pixels which belong to high luminance areas. The version is used is without colour reconstruction.
- Melyan et al.'s operator [133] (**M**): is a straightforward method for creating specular highlights on HDR monitors. Two linear tone scales with thresholding are applied to LDR images, and discontinuities are solved using a Gaussian convolved threshold map.
- Wang et al.'s operator [207] (**W**): is a system for adding details in the over-exposed and under-exposed regions of an LDR image. This system transfers texture details from a correct exposed patch to over-exposed and under-exposed patches, followed by a morphological Gaussian expansion.
- Rempel et al.'s operator [171] (**R**): is a global linear expansion, where an expand map reconstructs over-exposed areas. The expand map is based on thresholding and Gaussian filtering.

For a more in depth review of these see Chapter 3. All the iTMOs were implemented by the author. Akyüz's operator [7] and Melyan's operator [133] are very straightforward to implement. However, Rempel's operator [171] and Wang's operator [207] are quite complex to implement. Authors of these two operators were contacted to gain access to the original implementations, but they were not available for public use because they were developed inside private companies. Therefore, an attempt was made to recreate the original implementation by following the details outlined in the original papers very closely.

6.2 Validation using Psychophysical Experiments

This section presents the psychophysical study on iTMOs. The study is inspired by the one on TMOs by Ledda et al. [108], which employed pairwise comparisons as well.

6.2.1 Experimental Framework

In HDR imaging there are three popular methods used for psychophysical experiments which are: rating [51, 102, 103, 225, 226, 25, 26], ranking [11, 25, 7] and paired comparisons [102, 103, 108, 9]. These methods implicate a necessity for a subject/participant to make simple judgements on selection, grouping, sorting and ordering of stimuli properties such as preference, similarity, etc.

In this study paired comparisons with a reference image was chosen as the methodology. Rating and ranking have some drawbacks compared to paired comparisons.

Rating has three main problems: range effects, frequency effects, and distribution effects. In the first one, range scales can produce different results depending on subjects' perception of the relationship between the scale and stimuli. In the second one, scales are affected by how many times a subject sees stimuli. The subject needs prior training to gain confidence with scales and stimuli. In the last one, an unequal distribution of test stimuli along the range leads to different scale estimates. Therefore, this method needs an accurate definition of the range scale for each task with pilot studies and a training of subjects. Moreover, a large number of participants and stimuli is required to acquire valid and reliable data [93].

Ranking works well when differences are quite noticeable and fine judgement is not needed. Otherwise many paired comparisons are needed before establishing a ranking. Furthermore, it is not a very accurate method to measure performances because there is no mechanism to determine the difference between an image that comes immediately after another one in the rank. This means that it is not possible to determine whether two different ranks are statistically different or not.

For these reasons, a paired comparison was conducted where each inversely tone mapped image generated by an iTMO was compared with the inverse tone mapped image generated by the other iTMOs and the HDR reference. At any one time, the viewer was presented with three images displayed side by side on the Dolby DR-37P HDR display. The display was calibrated and has a luminance maximum peak of $3,000$ cd/m^2 and a minimum value of 0.015 cd/m^2. The reference was always displayed in the centre and the images produced by two iTMOs were presented on either side, see Figure 6.1. The task of the viewer was to observe the two

Figure 6.1: *The setup of experiments. On the left side a diagram showing the setup of the experiment. On the right side an example of a paired comparison using the Dolby DR-37P HDR display. Subjects were required to determine which image (left or right) was most similar to the reference (middle).*

images and to indicate which one appeared to be most similar to the reference with respect to a specific criteria which varied across the experiments as explained in Section 6.2.4 and Section 6.2.5.

The main advantage of gathered data using a paired comparison approach is that the degree of agreement amongst observers in their preferences can be calculated. Since each subject was instructed to assess the performance of all possible pairs, the main disadvantage was that a large number of images had to be displayed. On the other hand, this approach is more logical with the advantage of being straightforward. Although, adopting another approach might have been swifter, due to the limited number of inverse tone mapping algorithms published, such a paired comparison was a more suitable choice.

	A	**B**	**M**	**W**	**R**	Score a_i
A	-	0	0	0	0	0
B	1	-	0	1	1	3
M	1	1	-	0	0	2
W	0	0	1	-	0	1
R	1	0	1	1	-	3

Table 6.1: *An example of the a_{ij} preference matrix for a subject and an image for the comparisons of iTMOs.*

For each reference HDR scene, the total number of possible pairs is $(t(t-1)/2)$ where $t = 5$ is the number of iTMOs being tested. Therefore, each subject had to evaluate 10 pairs in order to asses all of the combinations for the five iTMOs. For any given pair, the subject had to select which image appeared most similar to the reference. The votes for all participants were then added into a single preference matrix.

Data collected in pair comparison for each image and participant is recorded in a matrix a_{ij}, where the value a_{ij} denotes the number of times iTMO$_i$ was preferred to iTMO$_j$. The score of a class or iTMO, a_i, is defined as the number of times that it was picked in a comparison with all others:

$$a_i = \sum_{j=1, j \neq i}^{t} a_{i,j} \tag{6.1}$$

For an example of the matrix and scores see Table 6.1.

Consistency and Agreement Coefficients

There are several methods to analyse paired data. An approach would be to use Thurstone's Law of Comparative Judgments [190] which is a measuring model for paired comparisons. Such a model however is more appropriate in the case where one would assume that there are perceptual differences between iTMOs. On the other hand, the chosen approach was to analyse the experimental data by primarily testing two "properties": the individual *consistency* of subjects and the overall *agreement* amongst them.

Consistency or transitivity is an important aspect to consider with paired comparisons. Due to the nature of the experiment a participant can make inconsistent choices when observing paired data. In the simpler case of evaluating three iTMOs for example, if the participant voted iTMO$_i$ to be closer to the reference than iTMO$_j$ and the latter closer than iTMO$_k$, then one would assume, by logic, iTMO$_i$ to be better than iTMO$_k$:

$$\mathrm{iTMO}_i \longrightarrow \mathrm{iTMO}_j \qquad\qquad \mathrm{iTMO}_j \longrightarrow \mathrm{iTMO}_k$$

then

$$\mathrm{iTMO}_i \longrightarrow \mathrm{iTMO}_k$$

where → is a closeness operator; operand on the left side is closer to the reference than the one on the right. In the case of a straightforward ranking, this is what would happen. On the other hand, paired comparisons allow for the case where iTMO$_k$ is voted to be better than iTMO$_i$ thus making an inconsistent choice:

$$\begin{tikzcd}
\text{iTMO}_i \arrow[dr] & \\
\text{iTMO}_k \arrow[u] & \text{iTMO}_j \arrow[l]
\end{tikzcd}$$

this situation is called *circular triad*. To determine such inconsistencies the *Kendall Coefficient of Consistency* can be used [180]. This is defined as follows:

$$\zeta = 1 - \frac{24c}{t(t^2-1)} \qquad \text{where } \zeta \in [0,1] \tag{6.2}$$

where t is the number of iTMOs and c is the number of inconsistencies per subject. ζ is calculated on a per-subject and per-image basis. If ζ is 1, then there are no inconsistencies and the data could be directly expressed as ranks. ζ will move towards zero as the number of inconsistencies increases. From the above, it may appear that straightforward ranking or rating might be more appropriate as it avoids transitivity issues, nevertheless asking an observer to rank an algorithm from first to last is not always natural behaviour. Measuring inconsistency is useful as it provides a clear indication of the similarity of the algorithms. If there are small differences in the iTMOs, high inconsistency would be expected as the task is hard for the observers.

The overall agreement amongst all participants in evaluating all iTMOs for each image can be judged via the Kendall and Babington-Smith's *Coefficient of Agreement* [180]. The coefficient is defined as:

$$u = \frac{2\Sigma}{\binom{s}{2}\binom{t}{2}} - 1 \qquad \text{where } u \in [-1,1] \tag{6.3}$$

where, s is the number of subjects. Σ is the sum of number of agreements between pairs of observers:

$$\Sigma = \sum_{i \neq j} \binom{a_{ij}}{2} \tag{6.4}$$

Such a coefficient is a suitable measure of association or correlation between a set of ranks. u represents the amount of agreement among the participants. It has a maximum value of 1 in the case of complete agreement, and -1 in the case of complete disagreement.

Tests of Significance

To reinforce the analysis of the data, a further investigation was conducted on the ratings from subjects via three statistical tests to search for significance. The idea is to firstly test if there was agreement amongst the participants. In the case of agreement, it is tested if iTMOs were perceived as different. Finally, if iTMOs are found to be perceived significantly differently; pairs of iTMOs, that give different perception, are searched. Subsequently, the set of all iTMOs is divided into groups of iTMOs, where iTMOs in same group do not give different perceptual differences and iTMOs in separate groups are perceived as different.

The first test has the goal to determine if the coefficient of agreement, u, is significantly different from the value that would be obtained if the comparisons were randomly made thus with no agreement amongst subjects. This formally corresponds to test the following null hypothesis, $H_0: \quad \mu = 0$, against $H_1: \quad \mu \neq 0$, where μ is the population coefficient of agreement, i.e. the coefficient if the whole population had rated the iTMOs. For a small number of observers (less than 6) and a small number of iTMOs (less than or equal to 8) the critical values of u are tabulated in statistics books, see for example Table U in [180]. On the other hand, for a large number of subjects, s, and iTMOs, t, the following test statistic is computed:

$$X^2 = \frac{t(t-1)(1+u(s-1))}{2} \tag{6.5}$$

which approximates a chi-square distribution with $\binom{t}{2}$ degrees of freedom (df). The critical values are tabulated in most statistical textbooks, see for example Table C in [180]. If the value of the test statistic X^2, Equation 6.5, is greater or equal to the tabulated critical values at a specific level of significance α the null hypothesis can be rejected and confirm that there is strong agreement amongst observers when observing the various algorithms compared to the reference. At $\alpha = .05$ and 10 df the critical value is 18.31.

After the agreement amongst subjects is concluded, the overall test of equality across iTMOs was performed. This test determines whether differences in scores are purely by chance or if indeed the differences are significant. This is formally equal to test the following null hypothesis:

$$H_0 : \quad \pi_{\mathrm{iTMO}_i} = \frac{1}{2} \qquad \forall \quad \mathrm{iTMO}_i \tag{6.6}$$

where π_{iTMO_i} is the unknown probability that the iTMO$_i$ is preferred to another iTMO. This is achieved by checking if there are differences in the scores a_i across the iTMOs. For each image, the standardised sum of squares of the scores S is computed as:

$$S = \frac{4}{st}\left(\sum_{i=1}^{t} a_i^2 - \frac{1}{4}ts^2(t-1)^2\right) \tag{6.7}$$

and compared with S_c, the upper α critical value of the χ^2 distribution with $t-1$ degrees of freedom. If the observed S value is greater than or equal to the corresponding critical value S_c, the null hypothesis H_0 is rejected. For example, from χ^2 tables, for $t = 5$ at 0.05 significance level, the critical value $S_c = 9.45$.

After a good agreement in rating across subjects and disagreement in ratings across iTMOs are concluded, pairs of algorithms that had significantly different ratings were investigated. Particularly if one algorithm is consistently preferred against the others. A natural way to do this is to calculate observed differences in ratings between two algorithms and to test them to see if these observed differences in scores are due to chance or if they are statistically significant. Since for t iTMOs there exists $t(t-1)/2$ pairs of iTMOs, multiple (i.e. $t(t-1)/2$) values of paired differences need to be calculated and hence a multiple comparison test is needed. In this case, the critical value, R, for determining the statistical significance, is defined as:

$$R = \frac{1}{2}W_{t,\alpha}\sqrt{st} + \frac{1}{4} \tag{6.8}$$

where $W_{t,\alpha}$ is the upper $(1-\alpha)$ quantile of the W_t distribution. At a significance level of 0.05 and $t = 5$ $W_{t,\alpha} = 3.86$, see Table 22 in [157]. This procedure is based on the range of the scores obtained by the five iTMOs and it is equivalent to Tukey's multiple comparison correction used in ANOVA [43]. Note that the value of R can be used to generate different groups of iTMOs based on significance.

6.2.2 Generation of Expanded Images for Experiments

The goal of experiments was to measure performances of different iTMOs against an HDR reference which was displayed on an HDR monitor. In order to achieve this, LDR images must be generated from the respective HDR counterparts and then expanded using iTMOs.

The first step was to create LDR images. The idea was to simulate the process that most consumer digital cameras and video-cameras do when they capture images or videos. Each reference HDR image was processed using an automatic exposure algorithm [111], and then it was scaled by an estimated exposure e. Subsequently, values of the image were clamped in the range $[0,1]$, and finally discretised at 8-bit by applying a gamma correction of 2.2. Note that the application of a known gamma simplifies the linearisation process, because only the inverse needs to be applied to linearise the signal.

Each operator was subsequently applied to the generated LDR images. Since each iTMO required different parameters, what follows are the settings used for these:

- Akyüz et al.'s operator [7] (**A**): has two parameters, γ and k. γ was set equal to 1, because it was shown, in the original paper, to be the preferred parameter. k, the maximum input intensity, equal to $L_{\text{w, Max}}$ which was the 95th percentile of the luminance of the original HDR image, thus avoiding outliers.

- Banterle et al.'s operator [16] (**B**): the maximum output luminance, $L'_{\max}$, was set equal to $L_{\text{w, Max}}$, and L_{white} equal to $2L_{\text{w, Max}}$ as suggested in the original paper. Following the original paper, the radius for the density estimation, r_t, was set equal to 16 and the threshold for the number of samples was set equal to 6.

- Meylan et al.'s operator [133] (**M**): values for thresholds were the same as the ones presented in the paper. $Y(\mathbf{max})$, the maximum value of the highlight, that was set equal to $L_{\text{w, Max}}$.

- Rempel et al.'s operator [171] (**R**): the total maximum luminance, was set equal to $L_{\text{w, Max}}$

instead of $4,800$ cd/m^2 as in the original paper. If the reference has a much lower or higher peak than $4,800$ cd/m^2 the generated image will appear much darker or brighter than the reference and lead to unfair comparisons.

- Wang et al.'s operator [207] (**W**): the same parameters presented in the original paper were used. In this case manual interaction was needed for generating images. All possible under-exposed and over-exposed regions were hallucinated, however in some cases it was not possible to find patches for hallucinating from these regions in the same image.

Note that tone mapped images were not used as starting LDR images for iTMOs, since all iTMOs have been designed and tested for content generated with traditional capturing devices, cameras and video-cameras, and not for tone mapped images. Moreover, an iTMO generally works in under-exposed and over-exposed areas of the image which, in tone mapped images, are typically well reproduced.

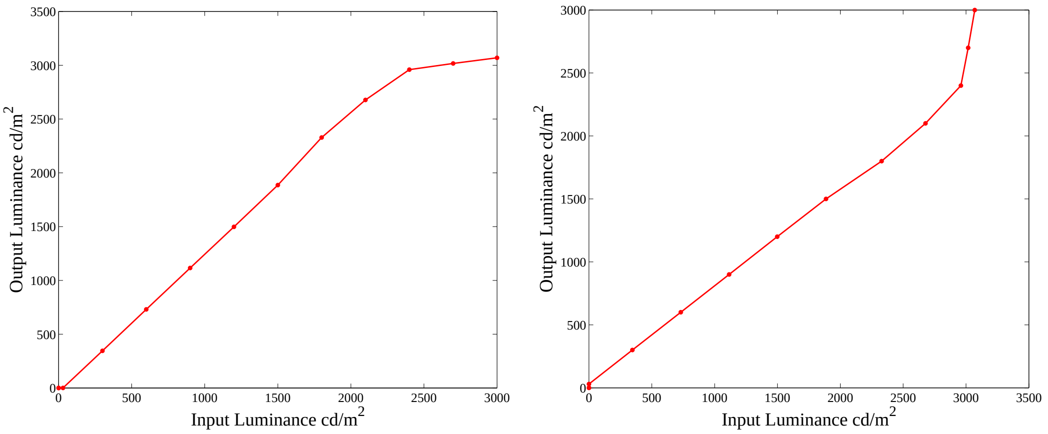
6.2.3 HDR Image Calibration for the HDR Monitor

The HDR images were visualised on the Dolby DR37P HDR monitor using Dolby's drivers, see Section 2.5 for an in depth review of the monitor. The drivers generate images appropriate for the LCD panel and LEDs in order to display a linear signal taking into account the gamma of the LCD panel and LEDs, see Trentacoste et al. [192] for more details about drivers.

The drivers perform a linearisation of the signal, which takes into account the non-linear nature of the LCD panel and LEDs. However, this process is partially realised. From measurements of white patches at different luminance values, it was observed that the generated signal by the drivers has two linear slopes, see Figure 6.2.a . This problem was solved by applying the inverse curve before the processing through the Dolby's drivers, Figure 6.2b.

Moreover, Dolby's drivers work only on the luminance channel without the implementation of a colour profile for the monitor. To solve this deficiency, each HDR image was firstly converted from RGB colour space to the XYZ one, sRGB colour space:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \mathbf{M}_{\text{RGB to XYZ}} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \qquad \mathbf{M}_{\text{RGB to XYZ}} = \begin{bmatrix} 0.4124 & 0.3576 & 0.1805 \\ 0.2126 & 0.7152 & 0.0722 \\ 0.0193 & 0.1192 & 0.9505 \end{bmatrix} \tag{6.9}$$



(a)

(b)

Figure 6.2: *Linearisation of the luminance signal in the Dolby DR37P HDR Monitor: a) the measurements of output luminance values at given input luminance. Note that the behaviour is not linear. There are two slopes. b) To correct the signal and have a linear one the inverse of the measurements in a) is applied to input values.*

then XYZ values are converted into the RGB colour space of the monitor:

$$\begin{bmatrix} R_{\mathrm{M}} \\ G_{\mathrm{M}} \\ B_{\mathrm{M}} \end{bmatrix} = \mathbf{M}_{\text{XYZ to mRGB}} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \qquad \mathbf{M}_{\text{XYZ to mRGB}} = \begin{bmatrix} 4.0955 & -2.1195 & -0.5164 \\ -1.0530 & 1.9575 & -0.0061 \\ 0.0921 & -0.2030 & 1.0007 \end{bmatrix} \tag{6.10}$$

where $[R_{\mathrm{M}}, G_{\mathrm{M}}, B_{\mathrm{M}}]^{\top}$ is a converted colour in the RGB colour space of the HDR Monitor. $\mathbf{M}_{\text{XYZ to mRGB}}$ was derived from measurements in Ruppertsberg et al.'s work [176].

6.2.4 Experiment 1: Image Visualisation

In this experiment, the goal was to investigate the quality of images generated by the five iTMOs compared to a reference HDR image. Three sub-experiments were conducted. The first was focused on the overall appearance of iTMOs with respect to the reference whereas in the second and third the focus was on under and over-exposed areas. iTMOs stretch the luminance channel especially in over-exposed and under-exposed regions of an LDR image, maintaining luminance values in well-exposed regions. Therefore, these regions need to be separately tested in order to have a better understanding of individual algorithms. In the first sub-experiment each participant was asked: *Which of the two images is the most similar to the* *reference?* In the second/third: *Observe the bright/dark areas only; which of the two images is the most similar to the reference?*

Stimuli Generation and Setup

A selection of eight common HDR images was used for this psychophysical study; these images represent indoor and outdoor scenes in various lighting conditions, from very dark to sunlight, see Figure 6.3. All of the HDR images selected for experiments were distributed around five orders of magnitude Table 6.2. This is within the contrast ratio limits of the Dolby DR-37P display. Then, inverse tone mapped images were generated applying the procedure in Section 6.2.2.

Figure 6.3: *The eight images used for Experiment 1. In the top row from left to right: Scene 1, Scene 2, Scene 3, and Scene 4. In the bottom row from left to right: Scene 5, Scene 6, Scene 7, and Scene 8. Scene 1-3 and 5-8 are courtesy of Ahmet Oğuz Akyüz [9]. Scene 4 is courtesy of Kimmo Roimela [174]. Note that a high resolution version of each image is shown in Appendix J.*

Image	Scene 1	Scene 2	Scene 3	Scene 4	Scene 5	Scene 6	Scene 7	Scene 8
Dynamic Range	3.2	3.8	5.5	5.4	3.2	5.3	4.4	3.2

Table 6.2: *The dynamic ranges of the HDR images used in the first experiment Figure 6.3.*

During the actual experiment a random sequence of all possible pair comparisons for all operators was generated. At any one time three images were presented to the participants: in the centre the HDR image and on either side an inverse tone mapped LDR image, chosen randomly, see Figure 6.1. Each pair combination was only presented once. The Dolby DR-37P display has a resolution of $1,920 \times 1,080$ pixels therefore, three images were resized in order to fit them on the monitor simultaneously. Image width was scaled to 600 pixels whilst maintaining the height-width ratio; in between each image a 50-pixel wide black band was inserted.

For each pair the participants had a maximum of 30 seconds to observe the three stimuli before answering. This was shown in a pilot experiment to be sufficient time to successfully complete the task. Since the total number of pairs was 80 (8 images × 10 pairs), each participant's experiment was split into two sessions of 20 minutes reducing the risk of observers getting tired or bored. A group of 24 naïve participants (18 men, 6 women) between 21 and 50 years old (mean age 28) with normal or corrected to normal vision took part in this experiment. The display was placed in a dark room minimising the effects of ambient light, and participants were seated one meter away from the monitor. Prior to the experiment, each subject was given five minutes to adapt to the environment and was presented with three trials in order to familiarise themselves with the experiment. The same group of participants took part in all three sub-experiments. The presentation order of the three sub-experiments was randomised.

Results: Overall Similarity Sub-experiment

The results for the first sub-experiment are shown in Table 6.3. The rows represent data obtained for each of the 8 scenes. In the first column of Table 6.5, the average consistency is tabulated. As can be seen, the consistency of participants was reasonably high implying that the task was sufficiently simple for the subjects. Such a result also indicates that the algorithms are different enough to be clearly assessed.

First, the agreement in rating amongst the participants was investigated. For this purpose the coefficient of agreement u, Equation 6.3, was calculated. The observed value of u for each scene can be found in Table 6.3. The higher the value the greater the agreement the between observers is. To determine the statistical significance of u the null hypothesis H_0, that there is no agreement amongst the raters, can be tested. This would imply that all iTMOs are perceptually equivalent causing difficulties in judging which image is most similar to the reference. In other words, it tests whether subjects made all their choices at random. To test the significance of u a large-sample approximation to the sampling distribution may be used. For each image the value of X^2, Equation 6.5, is compared to a critical value tabulated in statistics tables. As shown in

	Ave. Coeff. Cons. ς	Coeff. Agr. u	X^2	Sign. u	S	Sign. S	1st	2nd	3rd	4th	5th
Scene 1	0.700	0.210	58.333	$p < .05$	50.133	$p < .05$	B	R	W	A	M
Scene 2	0.508	0.059	23.667	$p < .05$	15.000	$p < .05$	A	B	R	W	M
Scene 3	0.792	0.476	119.500	$p < .05$	109.867	$p < .05$	W	B	R	M	A
Scene 4	0.917	0.650	159.500	$p < .05$	145.933	$p < .05$	B	W	R	M	A
Scene 5	0.858	0.149	44.333	$p < .05$	41.600	$p < .05$	B	A	W	R	M
Scene 6	0.867	0.392	100.167	$p < .05$	94.467	$p < .05$	B	M	W	R	A
Scene 7	0.692	0.263	70.500	$p < .05$	69.533	$p < .05$	M	B	R	W	A
Scene 8	0.500	0.058	23.333	$p < .05$	9.867	$p < .05$	R	B	W	A	M
Average	0.729	0.282	74.917	$p < .05$	67.050	$p < .05$	B	W	R	M	A
							531	434	411	295	249

Table 6.3: *The results for Experiment 1 Overall Similarity. The first column shows the average coefficient of consistency within subjects. The second the coefficient of agreement amongst subjects. In the third column the value of X^2 is presented which is approximately distributed as χ^2. From this value the significance u is determined, fourth column. The standardised sum of the squares of the scores S for testing of equality across iTMOs algorithms and its confidence are respectively in column five and six. Finally, the obtained ranks are presented. Any two iTMOs within the same circle cannot be considered perceptually different. The R value is equal to 21 for a single scene and equal to 60 for the Average.*

Table 6.3, for each scene the value of the test statistic u is larger than the 95% critical value, i.e. it is significant at the standard $\alpha = 0.05$ level. Typically, the agreement was much higher ($p < .001$). Therefore, it can be stated, with 95% (or even with 99.9%) confidence, that observers generally agree with each other when assessing the algorithms. Therefore, H_0 hypothesis, no agreements amongst the participants, is rejected for each scene. It can be concluded that the subjects generally agree with each other in assessing the algorithms.

The next step is to evaluate the agreement in rating across the iTMOs for each scene, using the standardised sum of squares S, Equation 6.7. To test for significance, S is compared to a critical value S_c for a desired significance level, $\alpha = 0.05$. As can be seen in Table 6.3, observed agreements across participants are highly significant ($p < .05$). Thus, the null hypothesis H_0, that the iTMOs are perceptually the same, can be rejected.

The previous test revealed that some iTMO algorithms are perceived differently from others. Now a natural question to ask is which pairs of algorithms are perceptually different. Therefore, the multiple comparison range test for each scene individually and for all scenes combined was performed. This was accomplished by comparing the pairs of algorithms via observed differences in ratings $a_i - a_j$ (defined in Equation 6.1). These differences are then compared with critical value $R = 21$ for individual scenes and with $R = 60$ for all scenes combined. These R values are visualised by grouping iTMOs in circles as shown in the tables. If the difference in scores a_i of any two iTMOs is larger than a critical value R, the algorithms are assigned to different circles and may be considered distinguishably different. Therefore, it can be concluded that iTMOs within the same circle are considered perceptually similar.

Although results are scene dependent, a reasonably clear pattern can be seen. iTMO **B** always ranks in the first group. It is also fairly clear that iTMOs **M** and even more so **A**, did not perform as well. One or the other was always ranked last for most scenes. The poor performance of **A** can be explained by the fact that the algorithm is a simple linear scale which, especially in dark HDR images, leads to overly bright images. Finally, the performance of iTMOs **R** and **W** were more or less the same exchanging position throughout the 8 test scenes.

Results: Bright and Dark Areas Sub-experiments

The results for the comparison of bright regions sub-experiment are presented in Table 6.5. As in the previous experiment, iTMO **B** was always ranked in the first group. The performance of iTMO **W** was also very positive being ranked in the first group for many of the scenes and in the second for the rest. iTMO **M** performed, once more, poorly. As in the overall comparison, the agreement u was always statistically significant. In this experiment the significance was even higher at the $\alpha = .001$ level. This result is also verified analysing the score differences which are highly significant ($p < .05$). In summary a strong evidence that for bright regions the perception differs across iTMOs was found, particularly **B** is the best iTMO for this sub-experiment.

	Ave. Coeff. Cons. ς	Coeff. Agr. u	X^2	Sign. u	S	Sign. S	1st	2nd	3rd	4th	5th
Scene 1	0.625	0.219	60.333	$p < .05$	55.200	$p < .05$	M	R	B	A	W
Scene 2	0.783	0.459	115.500	$p < .05$	106.867	$p < .05$	M	A	R	W	B
Scene 3	0.892	0.612	150.833	$p < .05$	137.533	$p < .05$	W	R	B	M	A
Scene 4	0.883	0.601	148.167	$p < .05$	140.200	$p < .05$	B	W	R	M	A
Scene 5	0.658	0.218	60.167	$p < .05$	54.200	$p < .05$	M	A	W	B	R
Scene 6	0.808	0.273	72.833	$p < .05$	64.867	$p < .05$	B	W	R	M	A
Scene 7	0.600	0.208	57.833	$p < .05$	52.667	$p < .05$	M	B	W	R	A
Scene 8	0.475	0.107	34.500	$p < .05$	11.867	$p < .05$	M	R	W	A	B
Average	0.716	0.337	87.521	$p < .05$	77.925	$p < .05$	M 477	B 411	R 405	W 382	A 245

Table 6.4: *The results for Experiment 1 Dark Areas. The R value is equal to 21 for a single scene and equal to 60 for the Average.*

The results of the comparison of dark regions sub-experiment are presented in Table 6.4. Interestingly, in this instance a complete reverse can be noticed in performances compared to the overall and bright areas sub-experiments. iTMO **M**, which performed relatively badly in the two previous sub-experiments, was ranked in the first group for the majority of scenes and was first on average. **A** performs poorly which may be related to the fact that linear scaling increases the luminance in the dark regions. The significance level of the data is the same as in the Bright Areas experiment. In conclusion the perceptually best algorithm for dark areas is **M**. There are not significant differences in perception for other algorithms, expect **A** which performed worst.

	Ave. Coeff. Cons. ς	Coeff. Agr. u	X^2	Sign. u	S	Sign. S	1st	2nd	3rd	4th	5th
Scene 1	0.808	0.578	143.000	$p < .05$	118.733	$p < .05$	B	W	A	R	M
Scene 2	0.717	0.328	85.333	$p < .05$	75.867	$p < .05$	B	A	W	R	M
Scene 3	0.708	0.268	71.667	$p < .05$	64.667	$p < .05$	B	W	R	M	A
Scene 4	0.775	0.257	69.000	$p < .05$	61.867	$p < .05$	B	W	M	R	A
Scene 5	0.858	0.374	96.000	$p < .05$	88.133	$p < .05$	B	W	A	R	M
Scene 6	0.833	0.399	101.833	$p < .05$	93.133	$p < .05$	M	B	A	R	W
Scene 7	0.625	0.193	54.333	$p < .05$	42.133	$p < .05$	M	B	A	W	R
Scene 8	0.642	0.228	62.333	$p < .05$	59.800	$p < .05$	B	R	W	A	M
Average	0.746	0.328	85.437	$p < .05$	75.542	$p < .05$	B 554	W 427	R 350	A 320	M 269

Table 6.5: *The results for Experiment 1 Bright Areas. The R value is equal to 21 for a single scene and equal to 60 for the Average.*

6.2.5 Experiment 2: Image Based Lighting

In a second experiment the performances of inverse tone mapping with regards to IBL was tested. Subjects had to make paired assessments of images generated using IBL with the five iTMOs and compare them to images generated using an HDR environment map. IBL consists of the re-lighting synthetic/real objects using an environment map which is achieved by the evaluation of a modified Rendering Equation [91], see Equation 2.80.

Since the appearance of an object depends strongly on the BRDF, different materials were tested. For example, a BRDF that represents a perfect specular mirror shows more details of an environment map, therefore differences between iTMOs might be perceptually more noticeable, because the integral in Equation 2.80 degenerates into a single weighted value, $L(\mathbf{x}, \omega')$. On the other hand, a pure diffuse material could mask these differences because the integral degenerates to an equal distribution on the hemisphere, that can be thought of as an averaging process. The performance of the five iTMOs was investigated for three classes of materials: pure diffuse, glossy, and pure specular reflective.

6.2.6 Stimuli Generation and Setup

Six HDR environment maps were selected as lighting data for the IBL experiment with different orders of magnitude, see Table 6.6. The environment maps are shown in Figure 6.4. Each HDR environment map was scaled and converted to a LDR one using the same technique presented in Section 6.2.2. The resulting LDR environment maps acted as input to the five iTMOs.

Figure 6.4: *Environment maps used for the Experiment 2 from the left to right: Saint Peter's Basilica (Scene 1), Eucalyptus Grove (Scene 2), Grace Cathedral (Scene 3) and Ennis (Scene 4), Pisa's square (Scene 5), and Campus (Scene 6). Scene 1-5 are courtesy of Paul Debevec [44].*

Image	Scene 1	Scene 2	Scene 3	Scene 4	Scene 5	Scene 6
Dynamic Range	5.4	3.7	7.1	4.6	3.7	3.2

Table 6.6: *The dynamic ranges of the HDR images used in the second experiment, see Figure 6.4. All used environment maps have a resolution of* $1,500 \times 1,500$ *pixels.*

For the experiment, a simple 3D scene consisting of a teapot was lit with the generated environment map images, see Figure 6.5. The teapot does not contain any high-frequency local geometry that could have disturbed the appearance of the illumination, such as the complex geometry which made it hard to distinguish illumination as shown in Ramanarayanan et al. [166]. The model is a standard teapot from the 3D modelling package Autodesk Maya [12], which is made of 6,300 triangles. Each teapot had a different material or BRDF assigned to it:

- **Pure Diffuse**: is a pure Lambertian BRDF with reflectance $\rho_d = [1,1,1]^\top$ in RGB colour space.

- **Glossy**: is a specular BRDF which spreads light towards an anisotropic direction. In the experiment, Ward's BRDF [215] was employed with specular reflectance $\rho_s = [0.9, 0.6, 0.15]^\top$ in RGB colour space and anisotropic direction $\alpha = [0.1, 0.25]^\top$. These settings were chosen to have a copper material appearance. The diffuse part of the full model, which is a pure Lambertian BRDF, was disabled in order to avoid influencing judgement.

Figure 6.5: *An example of IBL using the reference environment map for Scene 5: on the left side a render with the diffuse material, in the centre a rendering with a glossy material, and on the right side a rendering with a mirror material.*

- **Pure Mirror**: is a pure total reflective BRDF with reflectivity equal to $\rho_s = [1,1,1]^\top$ in RGB colour space.

A Monte-Carlo ray tracer rendered the images for each environment map, BRDF, and iTMO at the resolution of 600×600 with 4 samples per pixel, see Figure 6.5. Note that all images at exposure 0 are in Appendix K. With 10 comparisons for each environment map, the total image pairs to be evaluated were 60. A randomly generated sequence of such pairs was presented using the same approach as in the previous experiment. As in the first part of Experiment 1, the task was to select which image appeared to be closer to a reference image rendered with the original HDR environment map. For the IBL experiment, only overall performance was studied as distortions due to reflections would make it hard to distinguish dark and bright areas.

For each possible pair, as before each subject had a maximum of 30 seconds in which to give an answer. The experiment was split for each material (pure diffuse, glossy, pure specular). A group of 24 naïve participants (16 men, 8 women) between 19 and 56 years old (mean age 22) took part in this experiment. All exhibited normal or corrected to normal vision. Subjects were given five minutes to adapt to the environment and three test trials to familiarise themselves with the experiment.

6.2.7 Results: Diffuse Material

The results for the experiment with the diffuse material are shown in Table 6.7. The rows represent data obtained for each of the 6 environment maps. **B** performed better overall being placed in the first group on all occasions. **W**, **R** and **M** obtained similar overall results. **A** performed quite poorly overall and was placed last. As seen in Experiment 1, **A** performs badly because it is a simple linear scale. This linear process is equivalent to evaluating IBL using a normal LDR and subsequently scaling the resulting image. The outcome of the experiment was statistically significant ($p < .05$). Note that for IBL experiments $R = 21$ for each single image, and $R = 52$ for the Average.

	Ave. Coeff. Cons. ς	Coeff. Agr. u	X^2	Sign. u	S	Sign. S	1st	2nd	3rd	4th	5th
Scene 1	0.658	0.073	26.833	p < . 05	15.000	p < . 05	R	B	A	M	W
Scene 2	0.725	0.322	84.000	p < . 05	78.933	p < . 05	B	R	W	M	A
Scene 3	0.658	0.100	33.000	p < . 05	30.600	p < . 05	W	B	R	M	A
Scene 4	0.600	0.116	36.667	p < . 05	28.467	p < . 05	W	B	R	M	A
Scene 5	0.608	0.154	45.500	p < . 05	38.067	p < . 05	B	A	M	W	R
Scene 6	0.700	0.069	25.833	p < . 05	20.067	p < . 05	B	A	M	W	R
Average	0.658	0.139	41.972	p < . 05	35.189	p < . 05	B 355	W 305	R 282	M 280	A 218

Table 6.7: *The results for Experiment 2 Diffuse Material. The R value is equal to 21 for a single scene and equal to 52 for the Average.*

6.2.8 Results: Specular Materials

The results for the experiment with the glossy material are shown in Table 6.8. In this case they are similar results to the one obtained for diffuse material shown in Table 6.7.

	Ave. Coeff. Cons. ς	Coeff. Agr. u	X^2	Sign. u	S	Sign S	1st	2nd	3rd	4th	5th
Scene 1	0.733	0.166	48.167	p < . 05	41.933	p < . 05	B	R	M	W	A
Scene 2	0.742	0.225	61.833	p < . 05	60.267	p < . 05	R	W	B	M	A
Scene 3	0.542	0.015	13.500	p < . 05	11.667	p < . 05	M	B	R	W	A
Scene 4	0.608	0.063	24.500	p < . 05	12.667	p < . 05	B	R	W	M	A
Scene 5	0.617	0.073	26.833	p < . 05	19.533	p < . 05	B	A	M	W	R
Scene 6	0.633	0.076	27.500	p < . 05	18.133	p < . 05	B	W	A	M	R
Average	0.646	0.103	33.722	p < . 05	27.367	p < . 05	B 353	W 304	R 294	M 281	A 208

Table 6.8: *The results for Experiment 2 Glossy Material. The R value is equal to 21 for a single scene and equal to 52 for the Average.*

Finally the results for the perfect specular mirror material are shown in Table 6.9. In this experiment **B** again performed better. **W**, **R** and **M** had very similar performances. It is worth noting that **M** performed well for both specular materials for the Scene 3, possibly echoing the results from Experiment 1 where it performed best in darker areas.

	Ave. Coeff. Cons. ς	Coeff. Agr. u	X^2	Sign. u	S	Sign. S	1st	2nd	3rd	4th	5th
Scene 1	0.675	0.172	49.667	$p < .05$	45.400	$p < .05$	B	R	W	M	A
Scene 2	0.692	0.250	67.500	$p < .05$	59.000	$p < .05$	B	R	M	W	A
Scene 3	0.683	0.070	26.167	$p < .05$	17.667	$p < .05$	M	B	R	W	A
Scene 4	0.533	0.048	21.000	$p < .05$	12.667	$p < .05$	M	B	A	W	R
Scene 5	0.800	0.353	91.167	$p < .05$	84.667	$p < .05$	W	B	A	R	M
Scene 6	0.717	0.220	60.500	$p < .05$	52.200	$p < .05$	W	B	A	M	R
Average	0.683	0.186	52.667	$p < .05$	45.267	$p < .05$	B 381	W 315	M 275	R 274	A 195

Table 6.9: *The results for Experiment 2 Mirror Material. The R value is equal to 21 for a single scene and equal to 52 for the Average.*

6.2.9 Discussion

For the first experiment, the monotonically increasing functions **B**, **W** and **R** that enhance contrast non-linearly perform better overall and in many of the results are grouped together. The linear method **A**, and to a lesser extent **M**, perform worse overall, reflecting that for still images complex methods recreate HDR perceptually better.

For Experiment 2, the diffuse results show few differences. This is mostly due to the fact that rendering with IBL consists of the evaluation of an integral and during integration small details may be lost. This is less true for perfectly mirror-like or high glossy materials. However, in these cases details of the environment map reflected in the objects may be too small to be seen as is shown by the large groupings in the results. With certain scenes, for example Scene 4 and Scene 5 where the upper hemisphere of the environment map is uniformly well lit, the linear operators perform well. For more complex environment maps, the previously found ranking was reverted. Overall it is still clear that the operators that perform best, as with Experiment 1, are the non-linear operators.

Judging from the average results in the experiments, operator **B** performs better overall than the other operators, having been ranked in the top group for all sub-experiments except for Experiment 1, Dark Areas. This result may come at a cost since it is also the most expensive to compute. Nevertheless, when the most accurate results are required, this operator is currently the most likely to recreate most closely the original dynamic range of an image. **W** and **R** are commonly paired together in the average results. There is only one instance, Experiment 1, Bright Areas, when **W** outperforms **R** in the average results. **W** is not as expensive as **B** since the expansion is localised to require only the required under-exposed and over-exposed regions of the image. However, it requires a substantial amount of manual work. The results for **R** are very promising since it is fast enough to compute in real time using graphics hardware and while the experiment accounts only for still images, unlike **B** and **W**, it also adapts well to HDR videos. In the overall **M** ranks somewhere between, **W** and **R**, and **A**. In particular, for all the average IBL results it is grouped perceptually with **W** and **R**. Furthermore, most importantly it performed best in the experiments for dark regions. This conforms to the design of the operator, since it was primarily designed to enhance the specular highlights when expanding the dynamic range. These results are further strengthened by the operator's performance in the pure specular mirror material experiment were it came first in the two darkest scenes. Another advantage of **M** is that it is faster to compute than the other, non-linear, operators. Finally, **A** always ranks in the last group, except in the case of Experiment 1, Bright Areas. While **A** was designed to show that HDR images can be enhanced in a simple way to give a good HDR appearance as shown in Akyüz et al. [7], when attempting to recreate the original lost data this approach is insufficient. The advantages of this operator lie with its fast execution times and the possibility of using it with uniformly well-lit environment maps.

The experiments have shown that more advanced iTMOs, that cater for quantisation errors introduced during expansion of an LDR image, such as **B**, **R** and **W**, can perform better than simple techniques that apply single or multiple linear scale expansions, such as **A** and **M**. The more computationally expensive methods **B**, **R** and **W**, are better at recreating HDR than simple methods. Even if a linear scale can elicit an HDR experience in an observer, as shown in Akyüz et al. [7], it does not correctly reproduce the perception of the original HDR image.

6.3 Validation using Quality Metrics

In this section a study of iTMOs using computational metrics is presented. The study is divided in two parts. In the first one, quality is evaluated for still images. In the second part, flickering of iTMOs for videos.

6.3.1 Quality Evaluation for Still Images

Quality for still images was evaluated as for the psychophysical study in Section 6.2 for the two main applications: visualisation and IBL. Moreover, the same stimuli of Experiment 1 and Experiment 2 were employed in order to compare results and find possible connections.

Each inverse tone mapped image generated for Experiment 1 and Experiment 2 were compared against the HDR reference using an HDR metric. In HDR imaging there are few metrics that can deal with HDR values. These are: HDR Visual Difference Predictor (HDR-VDP) [127, 124], Root Mean Square Error (RMSE) in the $\log_2$[RGB] domain [223], and multi-exposure Peak Signal Noise Ratio (mPSNR). The last two metrics are statistical metrics that are mainly used for the evaluation of HDR compression algorithms. They are very sensitive metrics. For example, if an image and its shifted copy (few pixels) are compared, the error will be very high with these metrics. However, these images will be perceived as the same for the HVS. This problem can happen even in the case when pixel values are increased by a few intensity values. This is as in the case as inverse tone mapping, where reconstructed values are not exactly the same of the original image. Therefore, an HDR perceptual metric was employed. In particular, the used metric was HDR-VDP [127, 124] which is an extension for HDR images of the Visual Difference Predictor (VDP) [40]. The principle of VDP is to take into account limitations of the HVS rather than just compute upon physical values when comparing images. HDR-VDP outputs an image showing per-pixel false colouring highlighting the perceptual difference between images. Two values can be extracted from this resulting colour-coded image. The first is the percentage of different pixels detected with the probability of 0.75 ($P(X) \geq 0.75$). The second is the percentage of different pixels detected with the probability of 0.95 ($P(X) \geq 0.95$).

Recently, new metrics for comparing HDR images have been proposed such as the visual equivalence metric (VEP) [166] and dynamic range independent image quality assessment (DR-IIQA) [13] which are driven by the goal to improve or to propose an new alternative to HDR-VDP. VEP is not publicly available, including the needed psychophysical experiment data for running the model. An implementation of DR-IIQA is currently available via a web application with size limitations for images. Moreover, it is not practical for comparing medium/large data set of images, because the web application allows a comparison at the time with long uploading and processing times. Finally, VEP and DR-IIQA are not used in the computer graphics community which is still employing mostly HDR-VDP as the de facto standard perceptual image quality metric algorithm.

Results: Image Visualisation

The results of HDR-VDP comparisons for the image visualisation task are shown in Table 6.10. Each table entry records the HDR-VDP result for a given scene and operator in the case of $P(X) \geq 0.95$. **B** performed better on average than other operators. Moreover, there was a similar ranking as in Table 6.3. However, **W** performed worse than **A**, with two worst cases, Scene 6 and Scene 8, the error is 1.8-117 times more than **A**.

Rank	Scene 1	Scene 2	Scene 3	Scene 4	Scene 5	Scene 6	Scene 7	Scene 8	Average
1st	**B**	**R**	**M**	**M**	**R**	**B**	**B**	**B**	**B**
$P(X) \geq 0.95$	0.914	0.597	0.188	3.91	0.060	3.12	2.478	0.066	1.535
2nd	**A**	**B**	**B**	**B**	**B**	**M**	**M**	**R**	**R**
$P(X) \geq 0.95$	1.208	0.653	0.181	4.813	0.0594	3.537	2.656	0.073	1.887
3rd	**R**	**W**	**R**	**W**	**W**	**R**	**R**	**A**	**M**
$P(X) \geq 0.95$	1.265	0.812	0.198	4.996	0.075	4.256	3.07	0.149	2.138
4th	**W**	**M**	**W**	**R**	**M**	**A**	**W**	**M**	**A**
$P(X) \geq 0.95$	3.065	1.033	0.231	5.576	0.593	5.249	3.095	0.648	2.64
5th	**M**	**A**	**A**	**A**	**A**	**W**	**A**	**W**	**W**
$P(X) \geq 0.95$	4.542	1.299	2.826	5.928	1.264	9.814	3.196	17.51	4.95

Table 6.10: *The results of HDR-VDP comparisons for the Visualisation experiment. Values refers to the percentage of different pixels being detected with $P(X) \geq 0.95$ by the metric.*

Results: Image Based Lighting

The results of HDR-VDP comparisons for IBL are shown for the diffuse material in Table 6.11, for the glossy one in Table 6.12, and for the mirror material in Table 6.13. **B** performs on average better than other iTMOs for the diffuse and mirror material. Nevertheless, **M** performed better on average for the glossy material. As the for the image visualisation task **W** performed poorly, but better than **A** which performed as the last operator for all materials. Finally, the ranking relationships between **R**, **M** and **B** were kept for diffuse and mirror materials as in the psychophysical study shown in Table 6.7, Table 6.8, and Table 6.9. Note that, for the glossy case the average errors of **B**, **R**, and **M** are very close.

Rank	Scene 1	Scene 2	Scene 3	Scene 4	Scene 5	Scene 6	Average
1st	**M**	**M**	**B**	**B**	**B**	**B**	**B**
$P(X) \geq 0.95$	7.466	0.015	16.59	4.378	0.045	0.145	4.883
2nd	**B**	**B**	**R**	**R**	**W**	**W**	**R**
$P(X) \geq 0.95$	8.103	0.038	17.38	4.742	0.051	0.159	5.1
3rd	**R**	**A**	**M**	**M**	**R**	**R**	**M**
$P(X) \geq 0.95$	8.193	0.051	19.03	5.456	0.062	0.166	5.454
4th	**A**	**R**	**W**	**W**	**M**	**M**	**W**
$P(X) \geq 0.95$	8.399	0.0512	20.12	7.329	0.224	0.531	6.035
5th	**W**	**W**	**A**	**A**	**A**	**A**	**A**
$P(X) \geq 0.95$	8.444	0.111	20.42	13.35	0.222	0.531	7.161

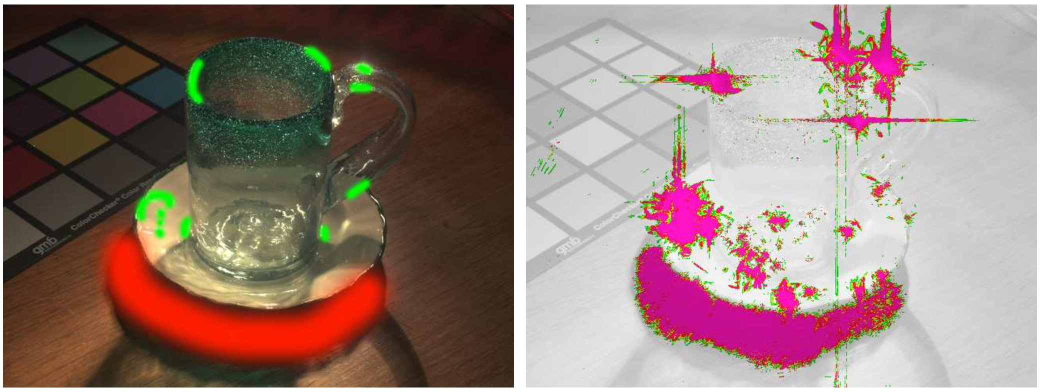
Table 6.11: *The results of HDR-VDP comparisons for IBL tasks for the Diffuse Material. Values refers to the percentage of different pixels being detected with $P(X) \geq 0.95$ by the metric.*

Rank	Scene 1	Scene 2	Scene 3	Scene 4	Scene 5	Scene 6	Average
1st	**M**	**M**	**B**	**M**	**B**	**B**	**M**
$P(X) \geq 0.95$	7.081	0.239	17.94	9.333	0.231	0.464	6.211
2nd	**B**	**A**	**R**	**R**	**W**	**W**	**B**
$P(X) \geq 0.95$	7.782	0.403	18.37	9.948	0.293	0.515	6.234
3rd	**R**	**B**	**M**	**W**	**R**	**R**	**R**
$P(X) \geq 0.95$	8.183	0.403	18.3	10.27	0.451	0.697	6.35
4th	**A**	**R**	**W**	**B**	**A**	**M**	**W**
$P(X) \geq 0.95$	8.414	0.453	19.21	10.58	0.769	1.538	6.6
5th	**W**	**W**	**A**	**A**	**M**	**A**	**A**
$P(X) \geq 0.95$	8.776	0.536	19.64	13.45	0.771	1.545	7.371

Table 6.12: *The results of HDR-VDP comparisons for IBL task for the Glossy Material. Values refers to the percentage of different pixels being detected with $P(X) \geq 0.95$ by the metric.*

Rank	Scene 1	Scene 2	Scene 3	Scene 4	Scene 5	Scene 6	Average
1st	**M**	**B**	**B**	**R**	**B**	**R**	**B**
$P(X) \geq 0.95$	14.27	0.901	32.18	8.634	1.572	3.56	10.73
2nd	**B**	**M**	**R**	**B**	**W**	**B**	**R**
$P(X) \geq 0.95$	17.27	0.9471	34.61	8.545	1.625	3.913	11.47
3rd	**R**	**R**	**M**	**M**	**R**	**W**	**M**
$P(X) \geq 0.95$	18.89	1.006	36.77	8.467	2.127	4.83	11.73
4th	**A**	**A**	**A**	**W**	**M**	**A**	**W**
$P(X) \geq 0.95$	19.09	1.066	39.65	10.32	4.729	5.202	13.06
5th	**W**	**W**	**W**	**A**	**A**	**M**	**A**
$P(X) \geq 0.95$	20.38	1.621	39.61	18.24	4.729	5.208	14.66

Table 6.13: *The results of HDR-VDP comparisons for IBL task for the Mirror Material. Values refers to the percentage of different pixels being detected with $P(X) \geq 0.95$ by the metric.*



(a) (b)

Figure 6.6: *An example of the concentration HDR-VDP error in Scene 6 for the visualisation task using* **W***: a) The starting LDR images and the brushes for hallucination. While covered pixel by the red brush were hallucinated using pixel from the wooden of the table, the others in green were hallucinated using pixels from the mug. b) The HDR-VDP output, note that error is mostly concentrated where brushes are. The original HDR image is courtesy of Ahmet Oğuz Akyüz [7].*

Discussion

For the visualisation application comparisons using HDR-VDP partially confirmed the trends that were shown in Section 6.2. **B** performed, overall as the best of the iTMOs. However, **W** had worse performance in this study than in the psychophysical one. On **W** the concentration of the error of the HDR-VDP output is mostly distributed in areas of the image which were hallucinated, see Figure 6.6. In these areas gradients and colours were transferred from different patches of the image, but they had similar features. For example, the under-exposed area near the mug was hallucinated from the well-exposed areas of the wooden table. Participants in the psychophysical study did not notice differences from the original pixels of the reference because that pattern looked similar, this effect is called visual masking [64]. However, HDR-VDP can not correctly determine the visual masking that happens in the HVS, and therefore the error was detected in the hallucinated areas.

Two main results have to be highlighted: the bad performance of **W**, and the overall good performance of **M** of the glossy material. The performances of **W** are due to hallucinated areas as in the case for the visualisation task. This is straightforward to show for the mirror material. Hence, a mirror visualises an image, so this case can be seen as a visualisation task and hallucinated areas are detected by the metric. For the diffuse and glossy materials the connection is more subtle. The hallucination of saturated lights introduced new structures into environment maps. These change how samples are distributed. On the other hand, when smooth gradients were transferred as in the case of Scene 5 and Scene 6 the operator performed better than in the other scenes. Regarding the performances of **M**, the first rank for the glossy material is because the operator is designed for reproducing highlights. These are usually important for good reproductions of a glossy material. However, as can be seen from all the IBL results tables the method failed to produce good results when a clear sky was present as in Scene 5 and Scene 6. This is because no clipped highlights were present in the environment map.

Note that the simple linear model, **A**, performed as the worst operator on average and in many scenes. This confirms a linear model is not suitable for IBL as shown in Section 6.2.5.

An interesting pattern can be seen from the HDR-VDP results for each scene and operator, see Figure 6.7. The pattern is an increase of the HDR-VDP error when the material increases in specularity. This is because the mirror material needs only a look-up in the environment map for the evaluation of Equation 2.80. This can produce a similar evaluation as for the visualisation application, although the surface is curved instead of being planar. On the other hand, the diffuse material needs more samples from the environment map for the evaluation of Equation 2.80, which are then averaged masking details. The glossy material is half-way between the two, less samples are required than a normal diffuse material, but more than a mirror material. Therefore, averaging is still happening but a low frequency environment map is reflected.

The error is higher in the IBL task than for the visualisation one, even if the area in the mirrored teapot is smaller than the one for visualisation. This is because initial environment maps were generated simulating the process of a real camera which employs automatic exposure. Automatic exposure does not well-expose pixels in the light source, which are needed for accurate IBL. It tries to minimise under-exposed and over-exposed pixels, saturing lights most of times. The use of automatic exposure is still a sound procedure for the generation of HDR environment maps. This is because most of existing the LDR environment maps/panoramas, for which inverse tone mapping is designed for, were captured whilst having over-exposed light sources.

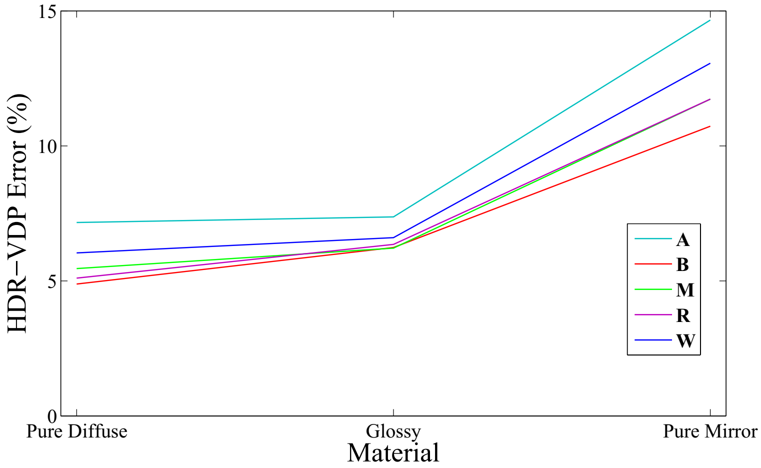


Figure 6.7: *This graph shows for the tested iTMOs that average HDR-VDP error* ($P(X) \geq 0.95$) *increases moving from a diffuse material to a mirror one.*

6.3.2 Temporal Coherence Evaluation for Videos

A few iTMOs have been proposed for the expansion of LDR videos. Moreover, a frame by frame expansion can introduce flickering in the original video. This is particularly evident in the case of **R**, which is designed for videos, because the expand map is based on thresholding. To test the quality of the temporally coherent expand map presented in Section 5.2, $\mathbf{B}_2$, a comparison between the two expand maps was conducted. In order to have expand maps in the same range, the one in $\mathbf{B}_2$ was mapped in the interval $[1,4]$ as in **R**.

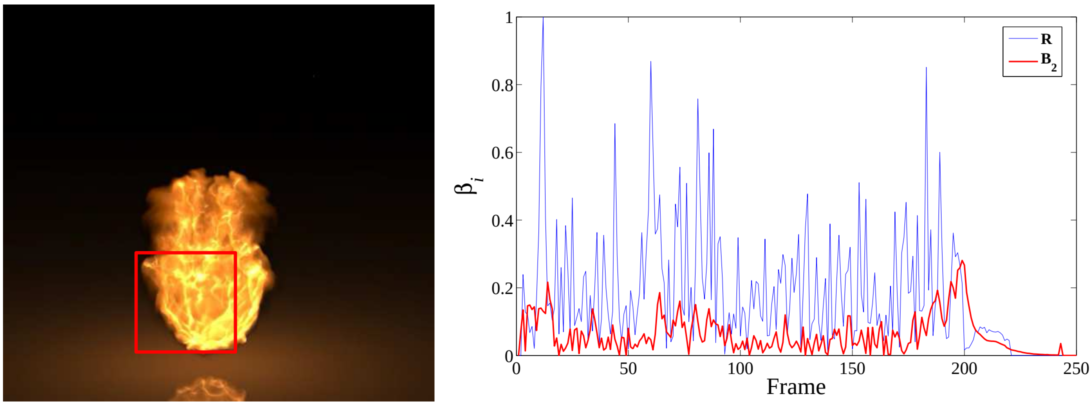
To have an estimate of flickering, the Brightness Flicking Metric (BFM) [201] was computed. BFM is a straightforward statistical metric which calculates the difference of the mean luminance channel between two frames:

$$\beta_i = \|\bar{L}_{i+1} - \bar{L}_i\| \tag{6.11}$$

where i is the current frame and $\bar{L}_i$ is the average of the luminance for the i-th frame or for a window in the frame. The metric is calculated for all frames of the sequence, which can be summarised in a graph. High variations in the graph mean that flickering can occur in the video sequence. To improve the estimate β_i was calculated in areas where fluctuations are frequent.

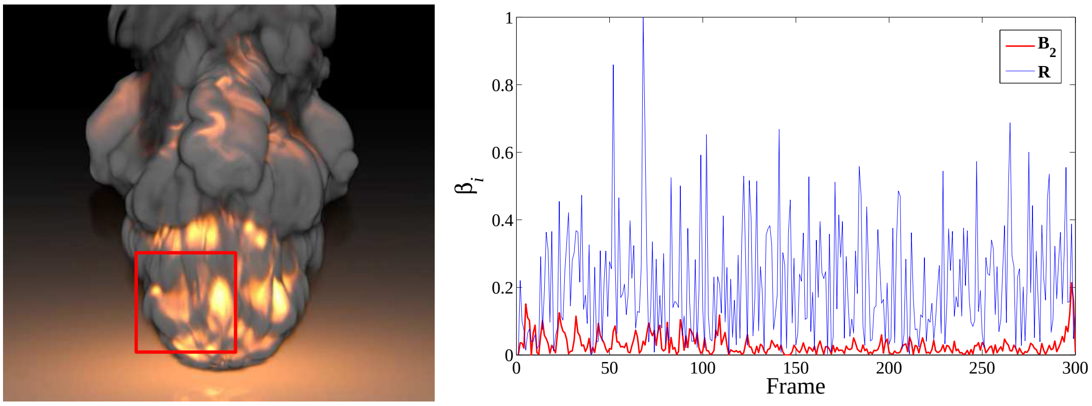
A challenging flickering test is to expand videos with fire, flames or smoke in it, because the animation is smooth but changes rapidly in these video sequences. The tested video sequences are `fireball.avi` and `fireball-smoke.avi` from Fedkiw's website [62], which are computer generated animations of fire and smoke. The results of BFM applied to expand maps of **R** and $\mathbf{B}_2$ are shown in Figure 6.8 and Figure 6.9. These two graphs show that $\mathbf{B}_2$ has less fluctuations and lower intensities than **R**, which means that it less prone to flickering artefacts.

Further tests were run for four shots from the video sequences `bbc-hd.mov` from BBC HD Gallery [19]. The results are shown in Figure 6.10, Figure 6.11, Figure 6.12, and Figure 6.13. As for the previous tests, $\mathbf{B}_2$ outperforms **R**. However, there are less peaks, because the motion is smoother than in `fireball.avi` and `fireball-smoke.avi`.



(a) (b)

Figure 6.8: *Flickering evaluation for* `fireball.avi` *sequence from Fedkiw's website [62]. On the left side the 58th frame of the sequence with a red square where* β_i *was calculated. On the right side the* β_i *values graph of the sequence. The* $\mathbf{B}_2$ *compared to* **R** *presents less fluctuations and lower intensities.*



(a) (b)

Figure 6.9: *Flickering evaluation for* `fireball-smoke.avi` *sequence from Fedkiw's website [62]. On the left side the 58th frame of the sequence with a red square where* β_i *was calculated. On the right side the* β_i *values graph of the sequence. The* $\mathbf{B}_2$ *compared to* **R** *presents less fluctuations and lower intensities.*

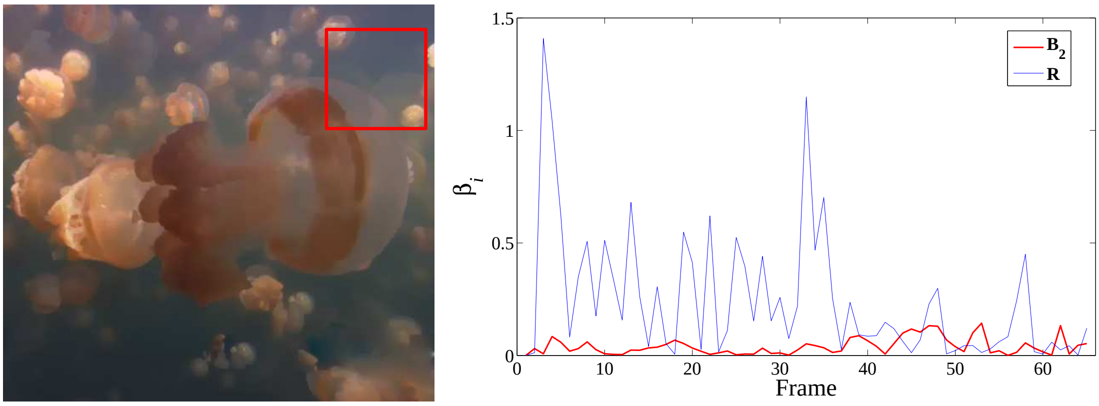


Figure 6.10: *Flickering evaluation for sequence 1 from* `bbc-hd.mov` *from BBC HD Gallery [19]. On the left side the first frame of the sequence with a red square where* β_i *was calculated. On the right side the* β_i *values graph of the sequence. The* $\mathbf{B}_2$ *compared to* $\mathbf{R}$ *presents less fluctuations and lower intensities.*

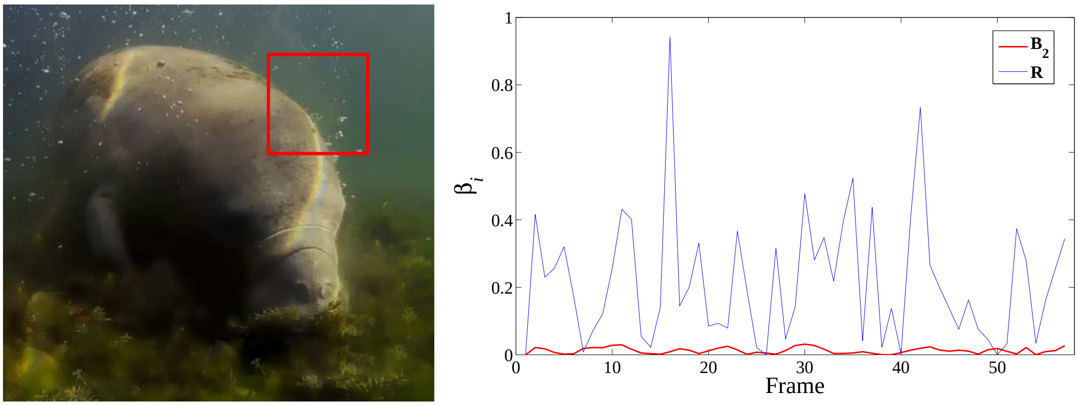


Figure 6.11: *Flickering evaluation for sequence 2 from* `bbc-hd.mov` *from BBC HD Gallery [19]. On the left side the first frame of the sequence with a red square where* β_i *was calculated. On the right side the* β_i *values graph of the sequence. The* $\mathbf{B}_2$ *compared to* $\mathbf{R}$ *presents less fluctuations and lower intensities.*

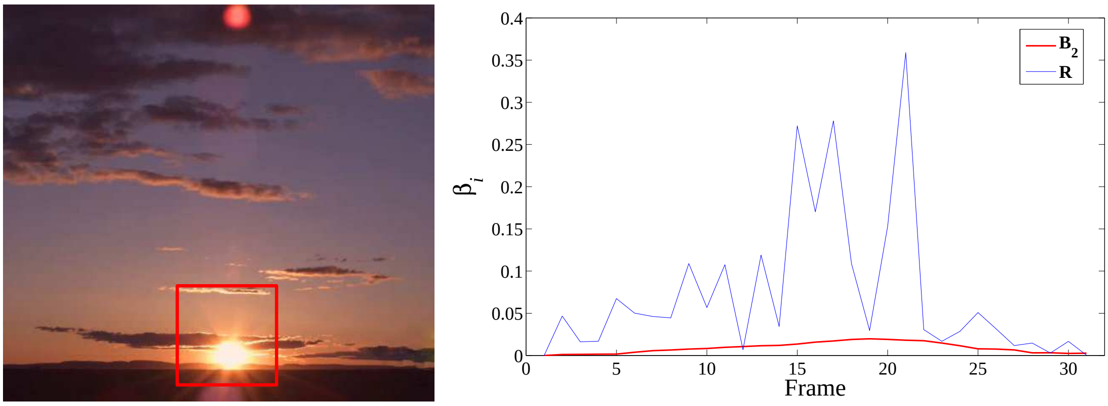


Figure 6.12: *Flickering evaluation for sequence 3 from* `bbc-hd.mov` *from BBC HD Gallery [19]. On the left side the first frame of the sequence with a red square where* β_i *was calculated. On the right side the* β_i *values graph of the sequence. The* $\mathbf{B}_2$ *compared to* $\mathbf{R}$ *presents less fluctuations and lower intensities.*

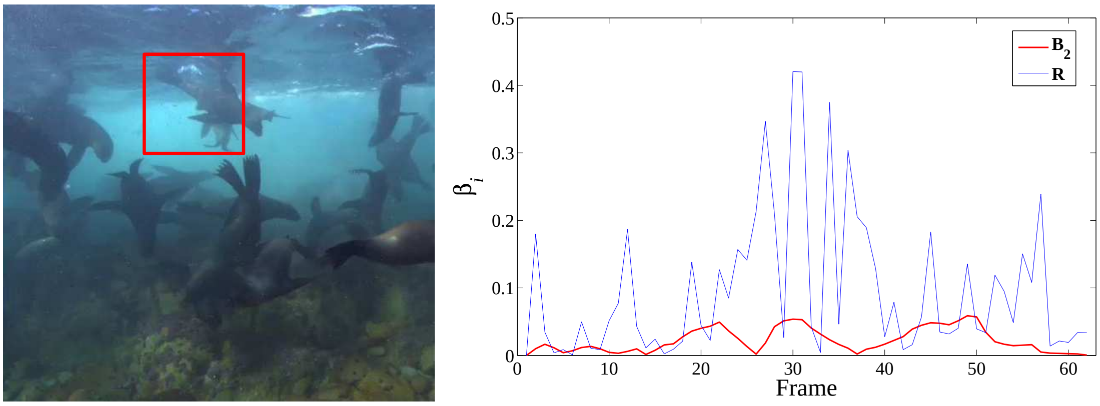


Figure 6.13: *Flickering evaluation for sequence 4 from* `bbc-hd.mov` *from BBC HD Gallery [19]. On the left side the first frame of the sequence with a red square where β_i was calculated. On the right side the β_i values graph of the sequence. The* $\mathbf{B}_2$ *compared to* **R** *presents less fluctuations and lower intensities.*

6.4 Summary

In this chapter, the quality of iTMOs was tested using a psychophysical study. This study proposed a methodology for the evaluation of iTMOs. Furthermore, it showed that non linear iTMOs with mechanism for reconstruction of missing content and reduction of noise have to be preferred to simple algorithms. The psychophysical study shared similarity to the results of the evaluation using HDR-VDP. Finally, the evaluation of flickering in videos showed that temporally coherent methods need to be employed when LDR videos are expanded.

Chapter 7

Inverse Tone Mapping Application: HDR Content Compression

HDR textures are highly valuable for creating content in real-time applications, off-line renders, and video-games. This is because IBL, Section 2.7, can dramatically improve rendering quality for reflections, refractions, definition of light sources, etc. However, a naïve representation of HDR textures needs 96 bpp which is a very high memory consumption. For instance, the upcoming video-game Rage [85] is going to have a single texture of a variable size from $16,384 \times 16,384$ up to $128,000 \times 128,000$ for a level, which is organised in sub-textures using an out-core technology called Megatexture [27]. Assuming a $16,384 \times 16,384$ texture in HDR using RGBE encoding, the uncompressed texture occupies 1 Gb of memory for a single level. This is a huge amount of memory for both storing and rendering. Moreover, this would be a problem even for an out-core technology especially at higher resolutions.

In this chapter a new compression method for HDR textures is presented [14]. This compression scheme can work on current GPUs with a bit rate of 8 bpp, which is achieved by using tone mapping for encoding and inverse tone mapping for decoding. The method uses the iTMO presented in Chapter 4. However, the expand map is not employed, because there is no need to reconstruct clipped areas, which are absent in tone mapped images.

7.1 General Compression Framework

In this section the general framework for HDR texture compression is presented. Implementation details are described in Section 7.2.

The main concept of the framework is to reduce the dynamic range of a HDR texture using a TMO, and to exploit LDR texture compression schemes to encode a tone mapped texture. Then, the HDR texture is reconstructed applying inverse tone mapping to the encoded tone mapped texture. The scheme is divided into two parts: encoder and decoder.

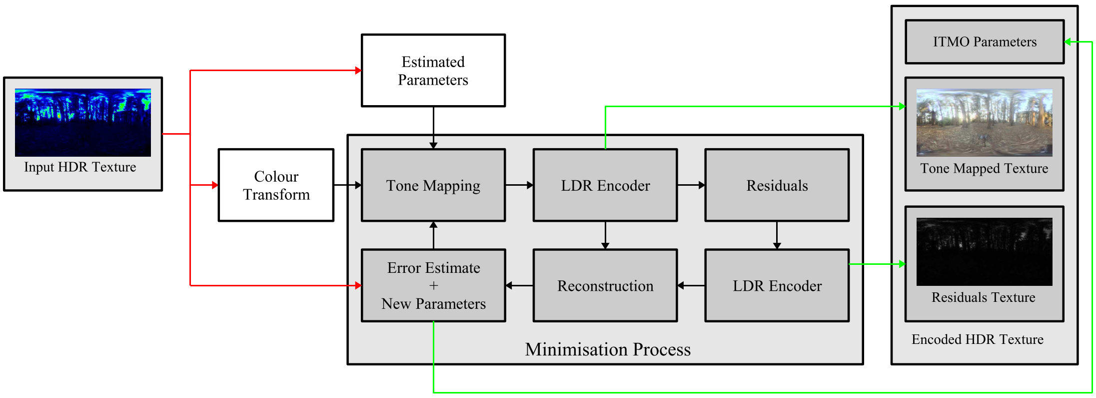


Figure 7.1: *The pipeline for encoding HDR textures. The image is tone mapped and stored using a LDR encoder. Then, residuals are computed to improve quality. A feedback loop determines the best parameters for tone mapping. This minimises the error between the original HDR texture and the reconstructed one.*

The encoder generates compressed textures starting from uncompressed ones, see Figure 7.1 for the full pipeline. Firstly, the texture is converted in an appropriate colour space based on the properties of the used compression scheme for LDR textures. Secondly, it is initially tone mapped using a TMO, where parameters are estimated based on a heuristic depending on the TMO. Then, the tone mapped texture is stored with a codec for LDR textures. Residuals are calculated from this encoded LDR texture and stored with the same LDR codec. At this point, the error between the original HDR texture and the reconstructed one is calculated. Finally, new parameters are estimated for the next iteration of this process. If a local minimum is reached the process stops, saving the current encoded tone mapped texture, residuals, and parameters for the TMO.

The decoder reconstructs HDR pixels when they are needed to be fetched in a shader, see Figure 7.2 for the full pipeline. The first step is to decode the compressed tone mapped texture.

Then, luminance is calculated, and inverse tone mapping is applied to it using the stored parameters. Subsequently, residuals are added to the expanded luminance. Finally, the chrominance channels are recombined to luminance to obtain the final HDR pixel.

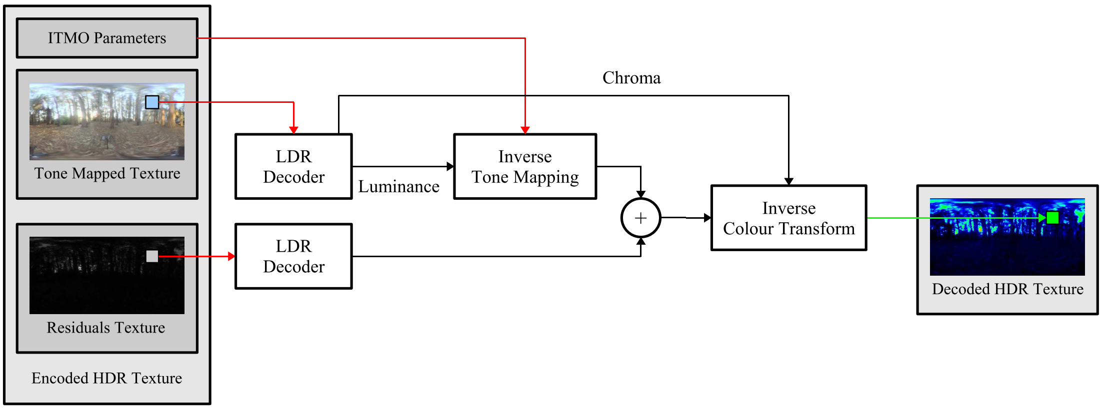


Figure 7.2: *The pipeline for the decoding of a HDR Texture. The needed pixel is fetched from the decoded tone mapped imaged, and expanded using inverse tone mapping. Finally, residuals are added to luminance, which is combined with chrominance channels to obtain the final HDR pixel.*

7.2 Implementation of the Framework

The implementation details of the framework for compression and decompression are given in this section. The whole framework can be seen as constructed from the following blocks:

- **Tone Mapping and Inverse Tone Mapping Operators**: the former determines the transformation from a HDR domain to a LDR one, and how bits are allocated. The latter is computed by a simple inversion of the first, and it expands LDR pixels to HDR ones.
- **LDR codec**: is used for encoding and decoding tone mapped textures and residuals. It depends on current graphics hardware to ensure real-time performance.
- **Colour space**: allows to allocate more bits to important channels of the image as luminance. However, it depends greatly on the LDR codec and the TMO.
- **Residuals**: encode the difference between the reconstructed signal during expansion and the original one.
- **Minimisation**: is the process for determining the optimal parameters for the TMO/iTMO.

Some blocks are common to both in the encoding and decoding stages, therefore they are presented without a particular division between encoding and decoding stages. In addition to the implementation of these blocks, the decoding shader is described. It is written in only 4 lines, which allows it to be plugged into more complex applications with a small effort of integration.

7.2.1 Tone Mapping and Inverse Tone Mapping Operators

In the framework, a TMO and its inverse are needed to compress values during encoding, and to expand values during decoding. The photographic tone reproduction operator by Reinhard et al. [169] was chosen as TMO for the implementation. There are three reasons to use it. Firstly, it was shown that it has performed very well for compression of still images [214] and videos [125]. Secondly, the model has a small number of parameters that can be estimated easily without user intervention. Thirdly, it can be inverted analytically and its inverse does not require expensive power functions.

The forward and inverse versions of the photographic tone reproduction operator are recalled, because they are going to be used extensively in this chapter. Moreover, the inverse operator is not the same as in Equation 5.7, because harmonic average is used for both compression and expansion. The forward version is defined as:

$$\begin{cases} f(L_{\text{w}}(\mathbf{x})) = L_{\text{d}}(\mathbf{x}) = \frac{\alpha L_{\text{w}}(\mathbf{x})(\alpha L_{\text{w}}(\mathbf{x}) + L_{\text{w, H}} L_{\text{white}}^2)}{L_{\text{w, H}} L_{\text{white}}^2 (\alpha L_{\text{w}}(\mathbf{x}) + L_{\text{w, H}})} \\ \left[R_{\text{d}}(\mathbf{x}), G_{\text{d}}(\mathbf{x}), B_{\text{d}}(\mathbf{x})\right]^{\top} = \frac{L_{\text{d}}(\mathbf{x})}{L_{\text{w}}(\mathbf{x})} \left[R_{\text{w}}(\mathbf{x}), G_{\text{w}}(\mathbf{x}), B_{\text{w}}(\mathbf{x})\right]^{\top} \end{cases} \tag{7.1}$$

where L_{white} is the luminance white point, and α is the scale factor. While the inverse is given by:

$$\begin{cases} g(L_{\text{d}}(\mathbf{x})) = f^{-1}(L_{\text{d}}(\mathbf{x})) = L_{\text{w}}(\mathbf{x}) = \frac{L_{\text{white}}^2 L_{\text{w, H}}}{2\alpha} \left(L_{\text{d}}(\mathbf{x}) - 1 + \sqrt{(1 - L_{\text{d}}(\mathbf{x}))^2 + \frac{4 L_{\text{d}}(\mathbf{x})}{L_{\text{white}}^2}} \right) \\ \left[R_{\text{w}}(\mathbf{x}), G_{\text{w}}(\mathbf{x}), B_{\text{w}}(\mathbf{x})\right]^{\top} = \frac{L_{\text{w}}(\mathbf{x})}{L_{\text{d}}(\mathbf{x})} \left[R_{\text{d}}(\mathbf{x}), G_{\text{d}}(\mathbf{x}), B_{\text{d}}(\mathbf{x})\right]^{\top} \end{cases} \tag{7.2}$$

Note that there is also no saturation (inverse tone mapping) and desaturation (tone mapping) of colours, because they are usually applied to adjust image colours for visualisation purposes. Furthermore, Ward and Simmons [214] and Okuda and Adami [146] used them for compressing/expanding the gamut. This was needed because they were adopting a different colour space than RGB for the encoded tone mapped image. However, in this implementation of the framework the RGB colour space is used, and these reasons why are described in Section 7.2.3.

7.2.2 LDR Codec

The current de facto standard for LDR texture compression, as implemented on graphics hardware, is S3TC/DXTC [87], see Appendix B for more details. Therefore, it was chosen as the LDR texture encoder for the implementation of the framework in Section 7.1. This choice does not limit the generality of the framework, when a new standard will be available DXTC can be substituted.

DXTC works on 4×4 pixels blocks, where two 16-bit colours, c_0 and c_1, are calculated for each block (5 bits for the red channel, 6 bits for the green channel and 5 bits for the blue channel). A 2-bit index for each pixel is used for interpolating from c_0 and c_1 during reconstruction. In the framework two variants were employed: DXT1 and DXT5. DXT1 is very compact, 64 bits per 4×4 block, resulting in 4 bpp. DXT5 uses an additional 64 bits for encoding an 8-bit alpha channel resulting in 8 bpp. While DXT1 was used for a low quality version without residuals, DXT5 was chosen for encoding them providing high quality compression.

7.2.3 Colour Space

Colour spaces in compression algorithms are usually very important, because some limits of the HVS can be exploited by assigning more bits to more important colour channels. In HDR texture compression methods, such as in [206, 139, 174], more bits are assigned to the luminance channel rather than the chroma channels. However, RGB was adopted in the implementation of the framework for three reasons. Firstly, the photographic tone reproduction operator compresses only luminance, and colours are scaled as can be seen from Equation 7.1. Note that it is similar to the LUVW colour space proposed by Wang et al. [206]. The second reason is the decoding efficiency. A colour space or other colour spaces which concentrate information in the luminance channel, such as LogLuv [174] or Yxy, needs more computational operations during the decoding of a pixel. For example, assuming a linear colour space at least three dot products are needed at the end of decoding for converting from the colour space to the RGB one. These additional operations could be considered few, but in a complex application where many texture fetches are called a reduction of three instructions can be essential to ensure a smooth frame rate. Thirdly, S3TC does not allow a good mapping from a separated luminance and chrominance colour space to its RGB colour space. For example, if Yxy colour space is mapped on S3TC, Y will be encoded as G channel, 6 bits, and xy to R and B channels, 5 bits each. This is a very poor mapping for luminance, and it can produce higher quantisation artefacts than using RGB colour space, see Figure 7.3 for an example.

(a) (b) (c)

Figure 7.3: *An example of failure of the use of a luminance separated colour space in S3TC applied to Bottles HDR image: a) The full resolution image at exposure -3. b) A zoom of a), green square, of the compressed image using RGB colour space. c) A zoom of a), green square, of the compressed image using $Ir_Q b_Q$ colour space by Roimela et al. [174]. Note that there are more artefacts than in b). This is because S3TC does not allow free allocation onto bits of its coding.*

7.2.4 Residuals

To improve overall quality of the reconstructed HDR texture, residuals are employed. Residuals are the difference between the original HDR luminance and the reconstructed one expanding the tone mapped and S3TC encoded texture. They can be computed as:

$$\delta_a(\mathbf{x}) = L_{\text{w, ori}}(\mathbf{x}) - L_{\text{w, rec}}(\mathbf{x}) \qquad \delta_m(\mathbf{x}) = L_{\text{w, ori}}(\mathbf{x}) / L_{\text{w, rec}}(\mathbf{x}) \tag{7.3}$$

where $\delta_a(\mathbf{x})$ is the additive difference, $\delta_m(\mathbf{x})$ the multiplicative difference, $L_{\text{w, ori}}(\mathbf{x})$ the luminance of the original HDR texture and $L_{\text{w, rec}}(\mathbf{x})$ the luminance of the reconstructed HDR texture at pixel location $\mathbf{x}$. $L_{\text{w, rec}}(\mathbf{x})$ is computed as:

$$L_{\text{w, rec}}(\mathbf{x}) = g(D_{\text{S3TC}}(L_{\text{d, S3TC}}(\mathbf{x}))) \tag{7.4}$$

where $L_{\text{d, S3TC}}(\mathbf{x})$ is luminance encoded with S3TC, D_{S3TC} is a function that decodes S3TC compressed values. Residuals in the additive form, $\delta_a(\mathbf{x})$, are adopted in the implementation, because they do not introduce noticeable artefacts when compared to multiplicative ones, $\delta_m(\mathbf{x})$, and they can be filtered with bilinear and MIP-mapping. This was shown by Wang et al. [206].

Finally, residuals are quantised into the 8-bit alpha channel of the tone mapped texture, which is subsequently compressed by the DXT5 encoder. Note that residuals for the chrominance channels were not taken into account, because they needed an additional texture for storing them. Moreover, a TMO usually compresses only the luminance channel, keeping the colour information, so errors in colours due to quantisation during LDR encoding are lower than errors in the luminance channel.

7.2.5 Minimisation

The last step in the encoding part of the framework is the minimisation process which defines parameters of the TMO/iTMO. They need to be chosen to fit the data as much as possible in 8 bits for each colour channel. For example, if they are chosen wrongly, quantisation can appear before LDR encoding, see Figure 7.5. To achieve the best possible fit an error function is defined as:

$$\varepsilon(L_{\text{w}}) = \sum_{\mathbf{x}} \left\| \log(L_{\text{w}}(\mathbf{x})) - \log(g([f(L_{\text{w}}(\mathbf{x}))]_0^{255})) \right\|^2 \tag{7.5}$$

where $[\cdot]_0^{255}$ is the quantisation into 8 bits. ε is calculated in the logarithmic domain to be more robust to outliers. For example, few pixels with high luminance levels can potentially move parameters towards them in the linear domain, producing quantisation artefacts. Note that the LDR encoder is not present in Equation 7.5 to avoid excessive complexity in the formulation.

Minimisation can be uniquely determinate as in [146]. However, they did not take into account the quantisation process during conversion to LDR and LDR encoding, which does not have an analytical formulation. Moreover, it can not be determined for TMOs that can not be inverted, since inversion implies finding zeros in a non-linear function. For these reasons, Equation 7.5 is solved using numerical methods, in particular Levenberg-Marquadt minimisation [163] was adopted in the implementation of the minimisation process. This is an iterative algorithm which converges to a local minimum given a starting solution.

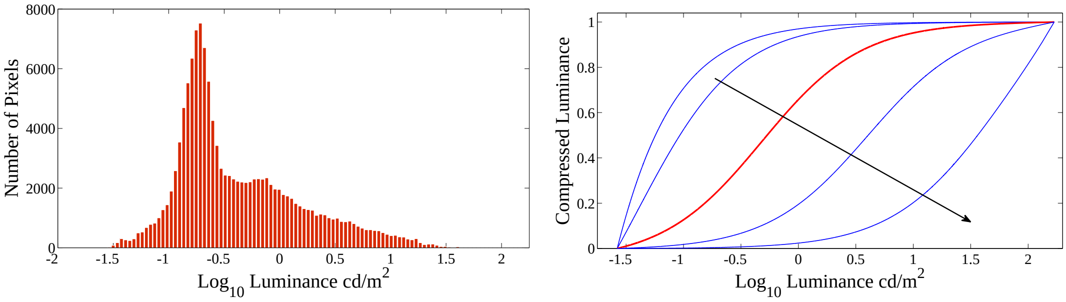


Figure 7.4: *An example of allocation varying α for the Eucalyptus's grove HDR image (Image 17 from Figure 7.11). On the left side the histogram of the image. On the right side the graph of various photographic tone reproduction operator's curve for $\alpha = \{100, 10, 1.203, 0.1, 0.01\}$. The arrow points to decreasing α values, from 100 to 0.01. The red curve is generated using Equation 7.6 based on image histogram. Note that the TMO allocates more space to pixels that have a large weight in the histogram.*

The photographic tone reproduction operator has two parameters α and L_{white}, which need to be minimised. These are initialised using automatic estimation based on heuristics. α scales HDR values and it can be seen as an exposure parameter. This scaling modifies how pixels in dark, medium and bright regions are compressed. For example, a $\alpha \approx L_{\text{w, Min}}$ keeps linearly values in very bright areas and compresses aggressively the other luminance values. Viceversa for $\alpha \approx L_{\text{w, Max}}$. A possible strategy is to determine a value for α which allows to keep linearly the more common values, because they contribute more to the appearance of the image. To achieve this, α is computed as the inverse of the mean luminance, see Figure 7.4 for an example. To include regions knowledge, a weighted mean based on the histogram is computed as:

$$\alpha' = \left(\frac{1}{N} \sum_{i=1}^{n_H} H(i) \text{Median}(i, L_{\text{w},i}(\mathbf{x})) \right)^{-1} \tag{7.6}$$

where α' is the estimated α for the first iteration of the minimisation, N is the number of pixels in the image, H is the histogram of the image, n_H is the number of bins in the histogram, and

$\text{Median}(i, L_{\text{w},i}(\mathbf{x}))$ calculates the median value in the i-th bin to be less prone to outliers of each bin. The histogram is calculated in the logarithm domain as defined by Ward et al. [107], where $n_H = 256$ provides an accurate precision for calculations.

L_{white} determines if a pixel is going to be burnt out or not. Hence if $L_{\text{x}}(\mathbf{w}) \geq L_{\text{white}}$ then $L_{\text{x}}(\mathbf{w}) = 1$. Since no pixels have to be burnt out, $L_{\text{w, Max}}$ is mapped to 1 in the tone mapped domain. This condition can be imposed as:

$$1 = \frac{\alpha' L_{\text{w, Max}}(\alpha' L_{\text{w, Max}} + L_{\text{w, H}} L'^{,2}_{\text{white}})}{L_{\text{w, H}} L'^{,2}_{\text{white}}(\alpha' L_{\text{w, Max}} + L_{\text{w, H}})} \tag{7.7}$$

which leads to:

$$L'_{\text{white}} = \frac{\alpha' L_{\text{w, Max}}}{L_{\text{w, H}}} \tag{7.8}$$

where L'_{white} is the estimated L_{white} for the first iteration of the minimisation. Note that L'_{white} depends on α' for the first iteration, but it does not depend on α during minimisation.

(a) (b) (c)

Figure 7.5: *An example of wrong parameters selection: a) The tone mapped image 15 from Figure 7.11 with a region of interest in green. b) A close-up of the region of interest after reconstruction using non estimated parameters for the TMO/iTMO. c) The region of interest after reconstruction using estimated parameters. As can be seen a non-automatic estimation of parameters may produce additional quantisation artefacts because not all 8 bits are used.*

7.2.6 The Shader for Decoding

The implementation of the decoding is straightforward. Only a few operations need to be performed: decoding of two LDR textures (tone mapped and the residuals textures), application of inverse tone mapping to the tone mapped texture, addition of residuals, an inverse colour transform. As shown in Section 7.2.3, the use of S3TC and the photographic tone reproduction operator bounds the implementation to the use of the RGB colour space, so there is no inverse colour transform. Moreover, the decoding of LDR textures is implemented in hardware, so there is only the need to call the hardware for this task.

```
1  sampler2D texTMO;
2  sampler1D texITMO;
3  const float3 LUMINANCE= float3(0.21,0.72,0.07);
4
5  float3 tex2DITMO(float2 texCoords)
6  {
7      //Fetch to the tone mapped texture
8      float4 tmoValue= tex2D(texTMO,texCoords);
9      //Compute Luminance
10     float L= dot(tmoValue.rgb,LUMINANCE);
11     //Fetch to the pre-computed iTMO
12     float Lw= tex1D(texITMO,L);
13     //Residuals addition, and combination with colours
14     return tmoValue.rgb*(Lw+texTMO.a)/L;
15 };
```

Listing 4: `tex2DITMO` *is the function which is used in a shader for decoding a compressed texture. This function takes 2 fetches to texture, 1 dot product, 1 multiplication, and 1 division.*

The decoding was implemented for programmable shaders, in particular the target shader language was HLSL for Direct3D9c [135]. However, the translation to other shading languages, such as Cg or GLSL, requires few modifications. The source code of the decoding function, `tex2DITMO`, is shown in Listing 4. The shader decodes the tone mapped texture `texTMO` (line 8), and computes the luminance `L` (line 9). Then, the iTMO in Equation 7.2 is applied to the tone mapped luminance (line 10) for obtaining HDR one, `Lw`. Note that to speedup on-the fly calculations, Equation 7.2 was pre-computed into a 1D texture, `texITMO`, with 256 half floating point numbers, for a total of 512 bytes. Finally, the residual value in the alpha channel `tmoValue.a` is added to `Lw` and colour are recombined, `tmoValue.rgb/L`, obtaining the decoded HDR value.

7.3 Analysis of the Error in the Texture Filtering

This section analyses the error of applying bilinear filtering to textures in Listing 4 before inverse tone mapping. This case is denoted as *pre-filtering* from now. The error needs to be determined for understanding if this approximation, which speeds-up the decoding four times,

(a) (b)

Figure 7.6: *An example of RGBE bilinear filtering for Image 17 from Figure 7.11: a) A correct bilinear filtered image using RGBE, filtering is applied after the RGBE decoding. b) A wrong bilinear filtered image using RGBE, where filtering is applied before RGBE decoding. Note that blocky edges appear, because exponents (which change rapidly across edges) were interpolated.*

is reasonable to have.

The used iTMO is not a linear function and this leads to errors during filtering operations. However, the error introduced does not produce noticeable artefacts as in the case of interpolation in RGBE, see Figure 7.6. This is due to two reasons. The first is that exponents are not interpolated such as in RGBE, but only base values of a function are filtered, see Figure 7.7 for an example. The second reason is that the iTMO is a monotonically increasing and continuous function. Therefore, an interpolated value $p \in [a,b]$ is still continuous without producing further edges.

(a) (b) (c)

Figure 7.7: *An example of bilinear up-sampling using the proposed method for Image 11 from Figure 7.11: a) Simulated bilinear filter in the shader. b) Up-sampling before applying inverse tone mapping function. c) Difference between a) and b) scaled 20 times.*

The error of using pre-filtering is mathematically defined as:

$$\varepsilon_R(A,B,C,D,x,y) = \|L_{\text{F, w}}(A,B,C,D,x,y) - \hat{L}_{\text{F, w}}(A,B,C,D,x,y)\| \tag{7.9}$$

where $L_{\text{F, w}}(A,B,C,D,x,y)$ is the corrected up-sampled luminance value, and $\hat{L}_{\text{F, w}}(A,B,C,D,x,y)$ is the pre-filtered luminance value, (A,B,C,D) is the filtering interval, and (x,y) are the coordinates of the sample in (A,B,C,D). $L_{\text{F, w}}(A,B,C,D,x,y)$ is defined as:

$$\begin{aligned} &L_{\text{F, w}}(A,B,C,D,x,y) = t_1 + y(t_2 - t_1) \\ &\qquad t_1 = g(L_{\text{d,A}}) + x(g(L_{\text{d,B}}) - g(L_{\text{d,A}})) \qquad t_2 = g(L_{\text{d,D}}) + x(g(L_{\text{d,C}}) - g(L_{\text{d,D}})) \end{aligned} \tag{7.10}$$

and $\hat{L}_{\text{F, w}}(A,B,C,D,x,y)$ is defined as:

$$\begin{aligned} &\hat{L}_{\text{F,w}}(A,B,C,D,x,y) = g(t_1 + y(t_2 - t_1)) \\ &\qquad t_1 = L_{\text{d,A}} + x(L_{\text{d,B}} - L_{\text{d,A}}) \qquad t_2 = L_{\text{d,D}} + x(L_{\text{d,C}} - L_{\text{d,D}}) \end{aligned} \tag{7.11}$$

However, relative error is more effective as a measure than an absolute one in Equation 7.9, because the dynamic range changes for each image. Relative error is defined as:

$$\varepsilon_R(A,B,C,D,x,y) = \frac{\|L_{\text{F, w}}(A,B,C,D,x,y) - \hat{L}_{\text{F, w}}(A,B,C,D,x,y)\|}{L_{\text{F, w}}(A,B,C,D,x,y)} \tag{7.12}$$

Note that the use of relative error simplifies the equation removing the term $\frac{L_{\text{white}}^2 L_{\text{w, H}}}{2\alpha}$. Therefore, ε_R depends only on the interpolation interval (A,B,C,D) and L_{white}. Furthermore, the dependency is strongly related to the difference between the minimum and maximum value in (A,B,C,D). This is due to the fact that an higher difference produces an higher error.

7.3.1 Analysis Results and Discussion

The relative error function, $\varepsilon_R(A,B,C,D,x,y)$, depends on L_{white} and the interpolation interval (A,B,C,D). Therefore, an analysis was conducted on HDR images in order to analyse the error in real conditions. A data set of 22 images representing different lighting condition was employed, see Figure 7.11. Each image was compressed with the proposed compression scheme.

Then, all possible 2×2 interpolation blocks/tiles were generated for each tone mapped image and classified in intervals based on the difference between the maximum and minimum element in the block, see Figure 7.8. Subsequently, the maximum and average ε_R errors were calculated for each block and averaged for each interval which they belonged to, see Figure 7.9.

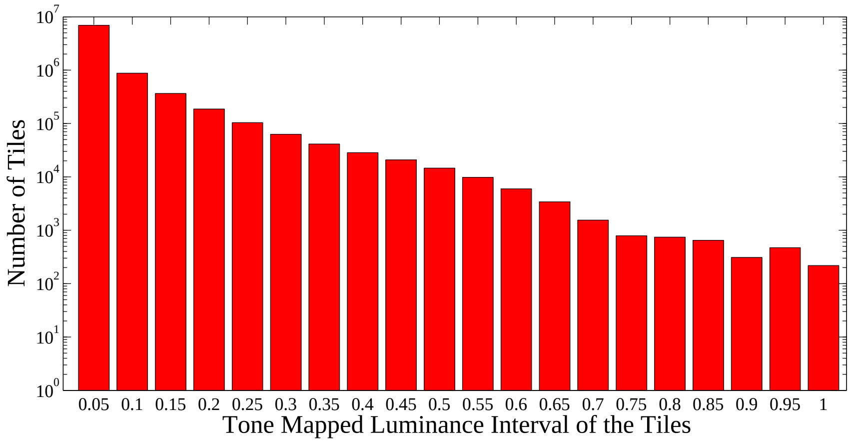


Figure 7.8: *The frequency of blocks interval in the* $\log_{10}$ *domain. Note that blocks in the difference interval* $[0, 0.15]$ *are more than* 93% *of blocks of the whole data set.*

As can be see from Figure 7.9, the error is on average around $3-5\%$ for the 88% of blocks reaching 10% for the 93% of blocks. However, the error for large intervals is quite significant reaching 82%. It has to be noted that only few blocks of the whole data set, 0.0002%, were matching this condition.

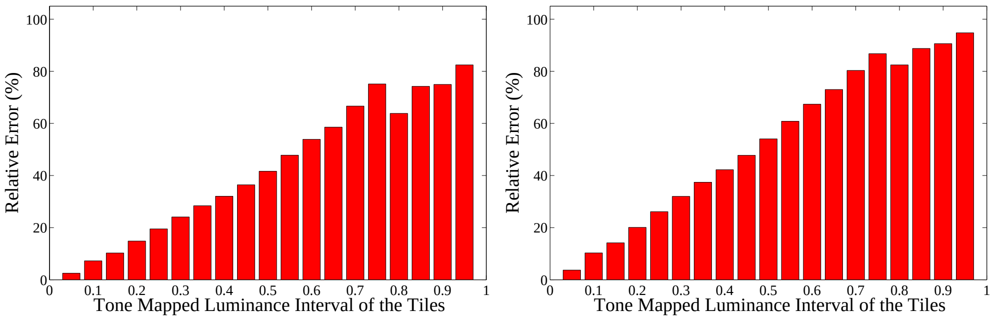


Figure 7.9: *The calculation of* ε_R *for each block in the data set and each interval: a) The average error. b) The maximum error.*

Furthermore, the large error can not be noticed by the HVS because no extra edges are introduced during interpolation, as the g function is a continuous function. A comparison between a corrected up-sampled image and a pre-filtered one was conducted using HDR-VDP showing that error is negligible for an image with the highest number of large intervals in the data set, see Figure 7.10.

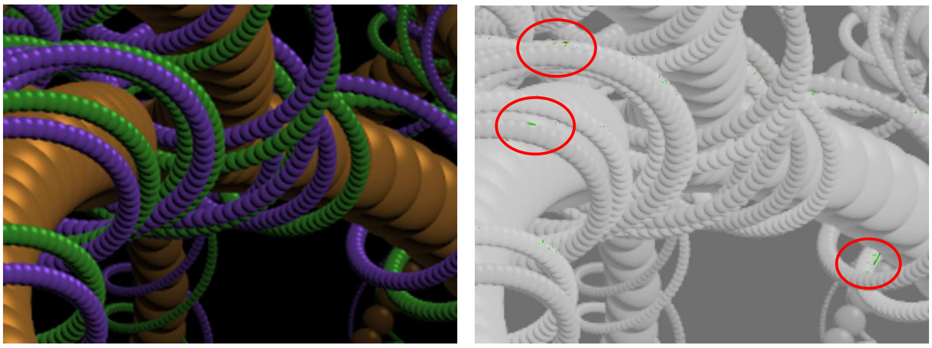


Figure 7.10: *The results of HDR-VDP between a corrected up-sampled image (4 times up-sampled image) and a pre-filtered one for Image 16 from the data set in Figure 7.11: a) Tone mapped image. b) HDR-VDP image. Note that error is very low ($P(X = 0.95) = 0.0285\%$), areas of high error are highlighted in red circles. Note that Image 16 has the highest number of large intervals in the data set.*

Regarding the up-sampling of residuals, Section 7.2.4 ,they can be filtered and added to the luminance channel without problems, because they are additive, so they are linear. A proof can be found in Wang et al. [206].

7.4 Compression Scheme Evaluation

This section presents how to measure performances of the implemented framework. In classic image compression, there are three main objectives that a compression scheme has to take into account: image quality, encoding/decoding computational complexity (depending on the applications), and compression rates. The implemented framework was analysed in terms of image quality, decoding speed, and compression rates. Encoding speed is not considered, because in many real-time 3D applications textures are produced before an application is distributed, and so there is no need for real-time encoding. Moreover, it is a rare to produce on the fly, and many times procedural techniques are employed which can help to avoid the need of compression [56].

7.4.1 Quality Metrics for Compression Evaluation

In order to evaluate the implementation of the texture compression scheme, see Section 7.2, common HDR metrics were used to validate results. These are: HDR-VDP [127, 124], a

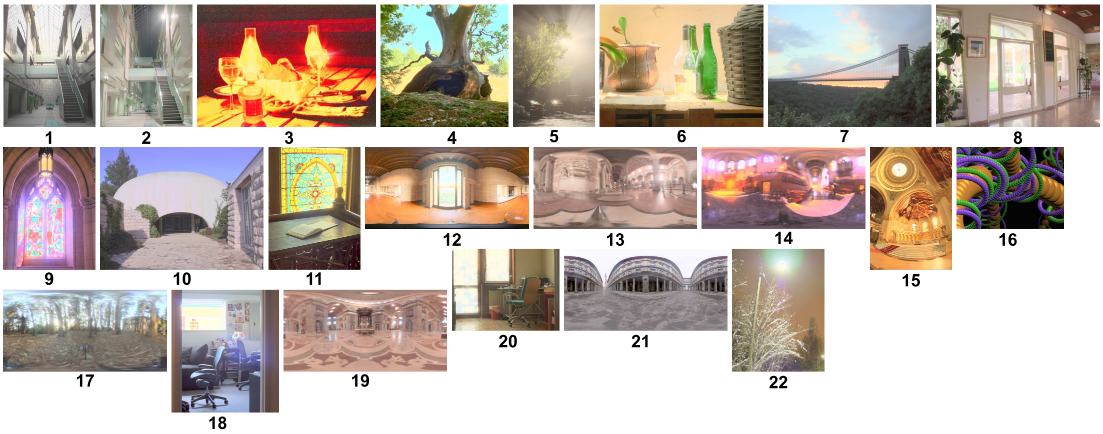


Figure 7.11: *The 22 HDR textures used in the compression experiments, tone mapped using the photographic tone reproduction operator [169] for visualisation. Images 1 and 2 are courtesy of Karol Myszkowski. Images 12, 13, 14, 15, 17, 19, and 21 are courtesy of Paul Debevec [47, 44]. Image 3, 4, and 11 are courtesy of Industrial of Light and Magic from OpenEXR SDK [86]. Images 7 is courtesy of Greg Ward [212]. Image 9 is courtesy of Max Lyons [121]. Image 8 and 10 are courtesy of Raan Fattal [61] .Image 16, 18 is courtesy of Jacob Munkberg [139]. Image 22 is courtesy of Kimmo Roimela [174].*

perceptual metric, Root Mean Squared Error (RMSE) in the $\log_2$[RGB] domain [223], and multi-exposure Peak Signal Noise Ratio (mPSNR), both of which are statistical metrics.

The RMSE in the $\log_2$[RGB] domain was proposed by Xu et al. [223], which is defined as follows:

$$RMSE(I,\hat{I}) = \sqrt{\frac{1}{n}\sum_{\mathbf{x}}\left(\log_2\frac{R(\mathbf{x})}{\hat{R}(\mathbf{x})}\right)^2 + \left(\log_2\frac{G(\mathbf{x})}{\hat{G}(\mathbf{x})}\right)^2 + \left(\log_2\frac{B(\mathbf{x})}{\hat{B}(\mathbf{x})}\right)^2} \tag{7.13}$$

where I is the reference image and (R,G,B) its red, green and blue channels, $\hat{I}$ the comparison image and $(\hat{R},\hat{G},\hat{B})$ its channels, n the number of pixels of the two images. A small RMSE value means that image $\hat{I}$ is close to the reference, zero means that they are the same, while a high value means that they are very different.

The mPSNR was introduced by Munkberg et al. [139], and it extends the peak signal-to-noise ratio (PSNR) to HDR images. This metric takes a series of exposures which are tone mapped using a simple gamma curve:

$$T(X,c) = \left[255(2^c X)^{\frac{1}{\gamma}}\right]_0^{255} \tag{7.14}$$

where c is the current f-stop, X is a colour channel, and $\gamma = 2.2$. Then the classic Mean Square Error (MSE) is computed:

$$MSE(I,\hat{I}) = \frac{1}{n \times p} \sum_{c=p_{\text{Min}}}^{p_{\text{Max}}} \sum_{\mathbf{x}} \left(\Delta R_c^2(\mathbf{x}) + \Delta G_c^2(\mathbf{x}) + \Delta B_c^2(\mathbf{x}) \right) \tag{7.15}$$

where p_{Min} and p_{Max} are respectively the minimum and maximum exposures, p is the number of used exposures, n is the number of pixels in the image, $\Delta R_c(\mathbf{x}) = T(R(\mathbf{x}),c) - T(\hat{R}(\mathbf{x}),c)$ for the red colour channel, and similarly for green and blue colour channels. Finally, the mPSNR is calculated using the standard formula for PSNR:

$$mPSNR(I,\hat{I}) = 10 \log_{10} \left(\frac{3 \times 255^2}{MSE(I,\hat{I})} \right) \tag{7.16}$$

The p_{Min} and p_{Max} can be automatically determined by taking all exposures from an under-exposed, mean luminance around 0.1, to an over-exposed image, mean luminance around 0.9. These two thresholds were chosen after an examination of p values presented in Munkeberg et al. [139].

7.4.2 Comparisons

The proposed implementation, Section 7.2, was compared against two other methods that are currently used in real-time applications, RGBE [209] and Wang et al. [206]. In addition to these methods, the method of Munkberg et al. [139] and Roimela et al. [174] were used for comparisons, although they are not currently supported by current OpenGL2/Direct3D9 and OpenGL3/Direct3D10 graphics hardware.

The encoder was implemented in Matlab R2007b using the Levenberg-Marquadt function implemented in the Optimisation toolbox. During the encoding stage, the local minimum was always reached starting with α' and L_{white} for the image set in Figure 7.11. Minimisation (Equation 7.5) takes less than three seconds using a $748 \times 1,128$ HDR texture. `texConv` from the Direct3D9c SDK was used as the S3TC encoder. The decoding was integrated into an existing Direct3D9 application. This step is straightforward and can be easily integrated into similar real-time graphics applications. Hence, only the shader in Listing 4 needs to be added to standard shaders. The encoding and decoding were tested on an Intel Pentium M 1.76 Ghz equipped with 1.5 GB of memory and a 256 MB GeForceGO 7300.

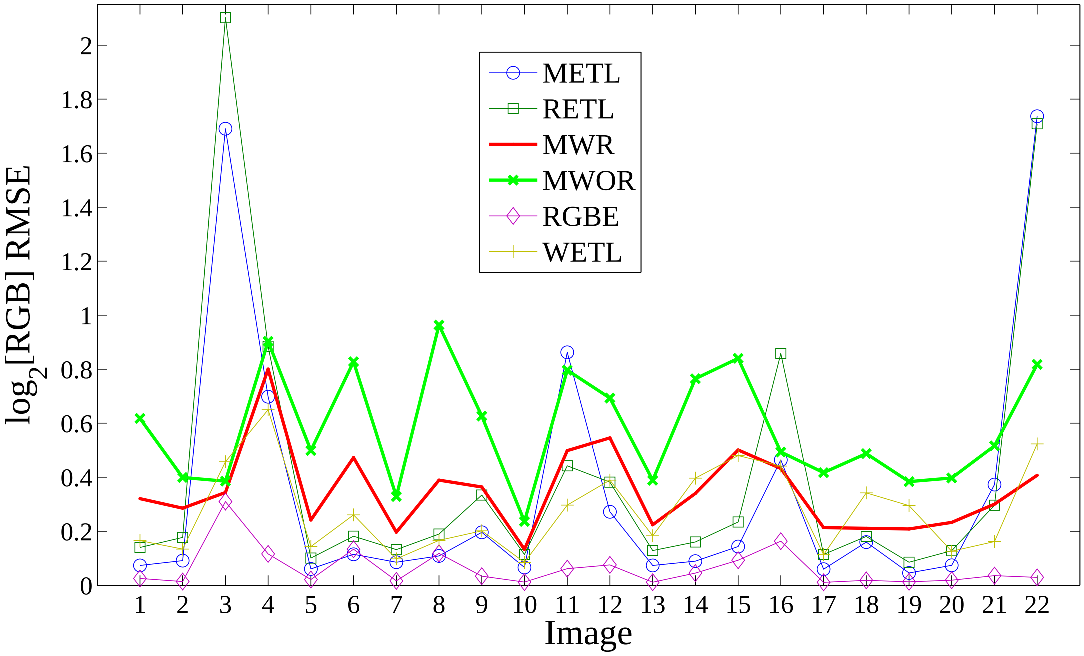


Figure 7.12: *The results of the comparisons using RMSE in the logarithm domain with the set in Figure 7.11 for RMSE.*

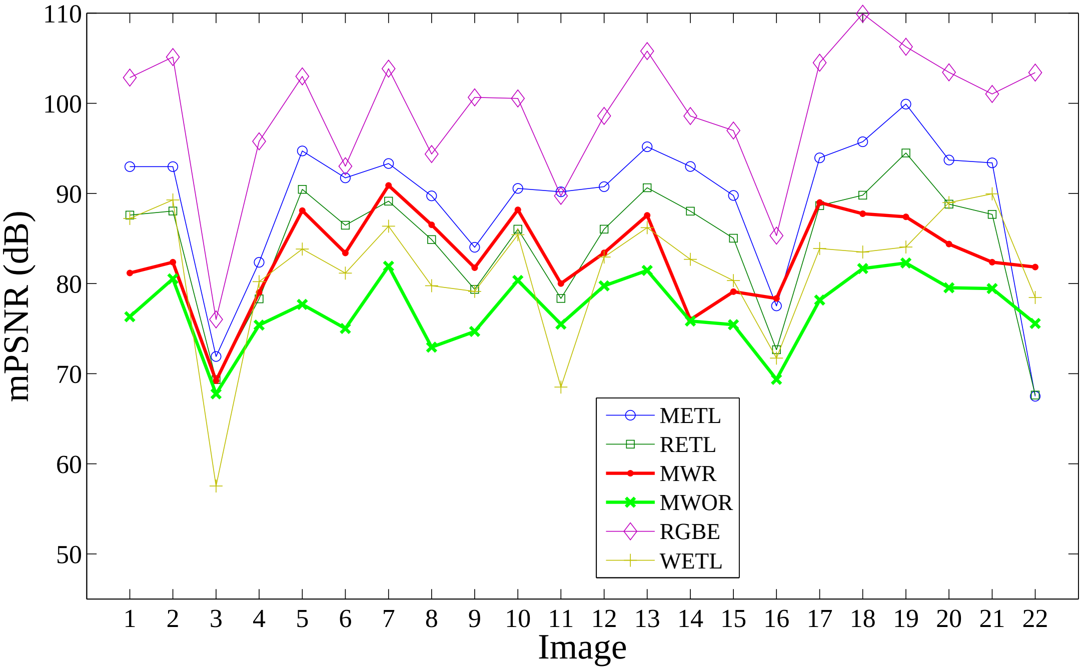


Figure 7.13: *The results of the comparisons using mPSNR with the set in Figure 7.11 for mPSNR.*

22 HDR textures, see Figure 7.11, were compressed with the six methods: RGBE, Munkberg et

al.'s method (METL), Roimela et al.'s method (RETL), Wang et al.'s method (WETL), and the proposed method with (MWR) and without residuals (MWOR). These textures were originally stored in the OpenEXR format (48 bpp). Figure 7.12, Figure 7.13 and Figure 7.14 show the values obtained for each picture using the metrics $\log_2$[RGB] RMSE, mPSNR and HDR-VDP. Table 7.1 summarises these results with the average value for each of the metrics over the total number of pictures. Furthermore, Table 1, presents a second value which serves to highlight the quality as a function of the compression. These results must be interpreted as less is better for HDR-VDP and $\log_2$[RGB] RMSE, while more is better for mPSNR.

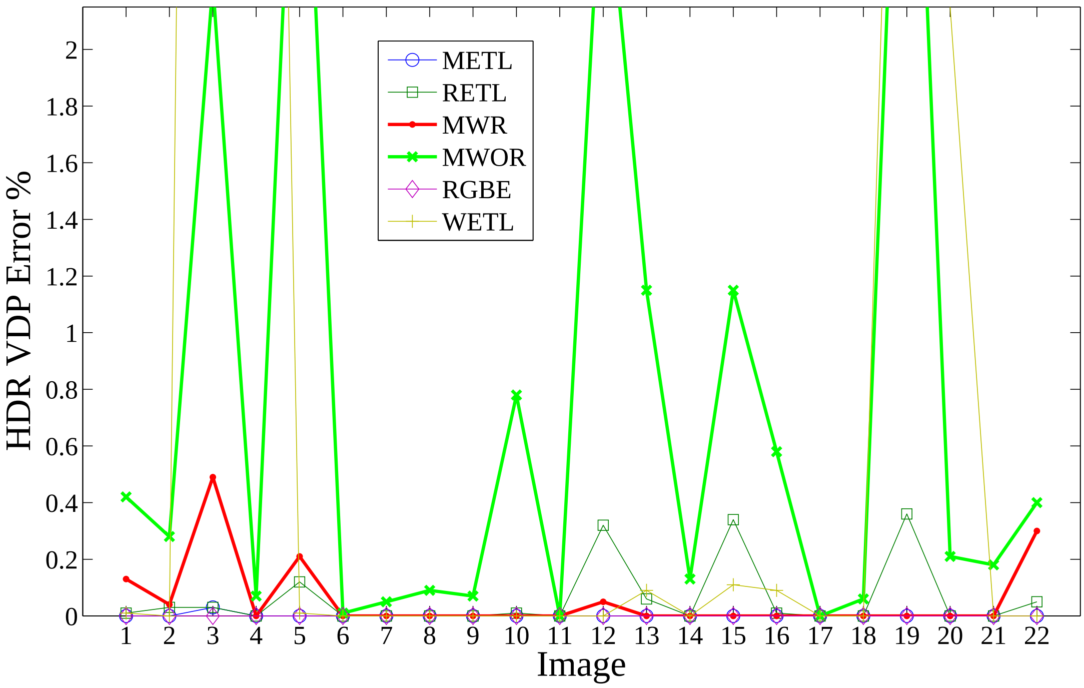


Figure 7.14: *The results of the comparisons using HDR-VDP with the set in Figure 7.11 for HDR-VDP with* $P(X) = 0.95$.

7.4.3 Discussion

RGBE performed as the best in terms of quality in all metrics from the results in Table 7.1. However, when compared as a function of quality over FPS or times the bit rate, it is best only in the case of the HDR-VDP metric. Furthermore, RGBE has problems when it comes to filtering and performance, RGBE bilinear filtering is implemented only on G80 graphics chips. The METL and RETL methods [139, 174] performed better on average than MWR and MWOR for straight out values, but less so as a function of quality. However, as mentioned

	RETL	METL	WETL	MWOR	MWR	RGBE*
Bits per pixel (Bpp)	8	8	16	4	8	32
Frames per seconds (FPS)	NA	NA	309	314	312	201
HDR-VDP average	0.0609	0.0014	1.333	0.815	0.1	0.0
HDR-VDP × Bpp	0.4872	0.0112	21.328	3.26	0.8	0.0
HDR-VDP / FPS	NA	NA	3.23e-3	2.59e-3	3.2e-4	0.0
$\log_2$[RGB] RMSE average	0.412	0.342	0.277	0.581	0.348	0.0623
$\log_2$[RGB] RMSE × Bpp	3.296	2.736	4.432	2.324	2.784	1.993
$\log_2$[RGB] RMSE / FPS	NA	NA	8.9e-4	1.85e-3	1.1e-3	3.1e-4
mPSNR average	84.43	89.31	81.12	77.12	83.11	99.03
mPSNR / Bpp	10.554	11.1638	5.07	19.28	10.3888	3.09
mPSNR × FPS	NA	NA	25066.1	24215.7	25930.32	19905.03

Table 7.1: *Metrics comparisons. This table shows the ratio of the average values of the used metrics and the bits per pixel, and as well for frames per second (FPS). FPS values were not calculated for MELT and RETL, because they can not be implemented using current graphics hardware.* $*$ *Note: RGBE is supported in hardware on a G80 series with filtering and it would run at 360 FPS on a G80 with the same speed as the used one in these tests.*

earlier, these methods need special hardware to run. Moreover, MWR and MWOR methods performed reasonably well, overall, they are better than WETL [206].

In conclusion, MWOR seems to offer the best compression to quality ratio in the cases of mPSNR and $\log_2$[RGB] RMSE, mainly due to the high compression ratio of 4 bpp. Note that WETL can suffer contouring artefacts, when the splitting axis can not find an optimal separation, as in the case of Saint Peter's Basilica HDR texture (Image 19 from Figure 7.11), see Figure 7.15 for an example.

Decoding speed is another meaningful parameter that has to be taken into account for real-time applications. In modern applications such as games many texture fetches are needed to calculate complex shaders and post-processing filters. To show the efficiency, the decoding times of MWOR, MWR, WETL, RGBE methods were timed using a simple Direct3D9 application where bilinear filtering and MIP-mapping were applied to textures. Real-time performance is summarised in Table 7.1. These values are timed from the simple application shown in Figure 7.15. MWOR and MWR obtained the fastest times, running respectively at 314 FPS and 312 FPS. RGBE was the slowest, at 201 FPS, due to the fact that bilinear filtering and MIP-mapping has to be emulated in a shader, otherwise artefacts at the edges are present in results. This is because exponents change rapidly at edges. WETL is nearly as fast as MWOR and MWR, 309 FPS, but it uses more operations: 3 multiplications, 3 additions, 1 dot product, 2 texture fetches compared to MWR which uses 1 multiplication, 1 addition, 1 dot product, 1 division

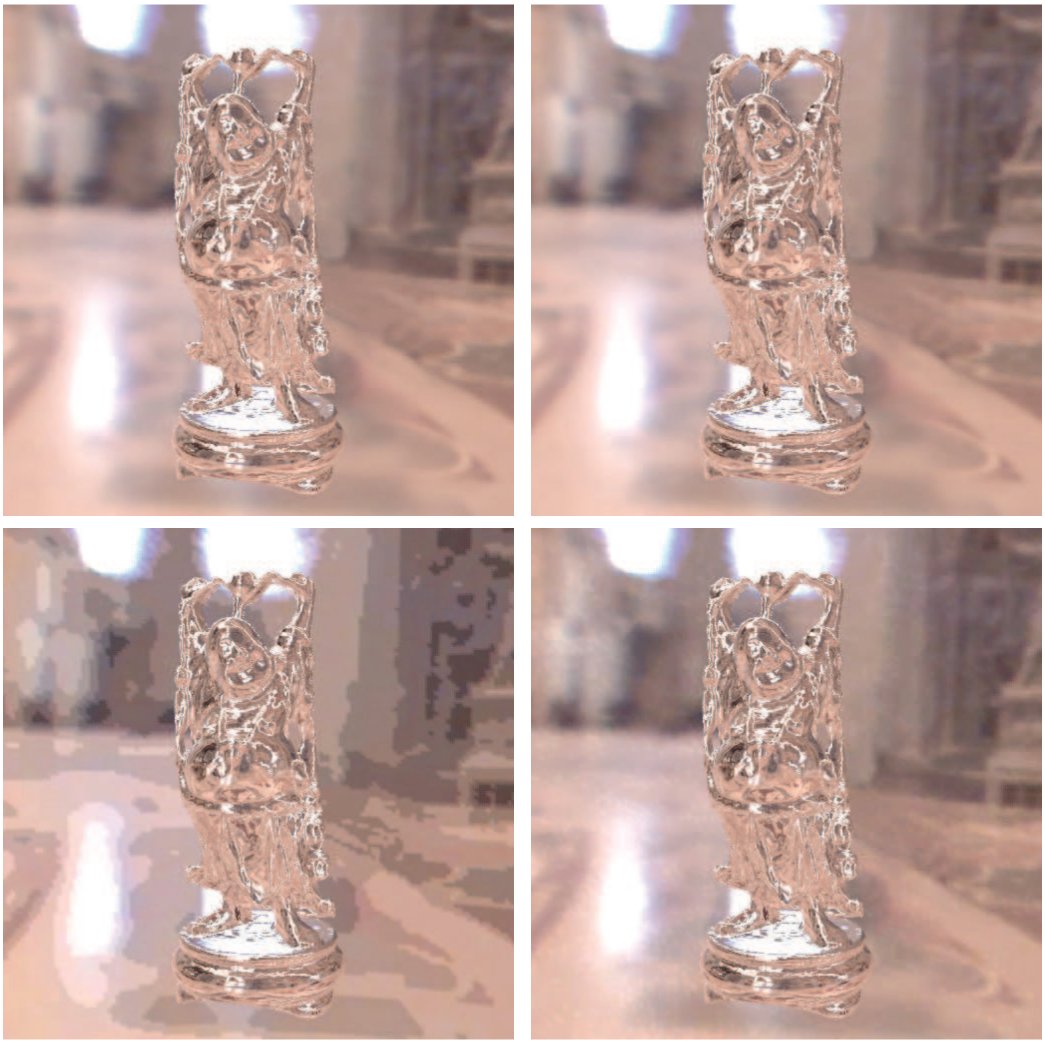
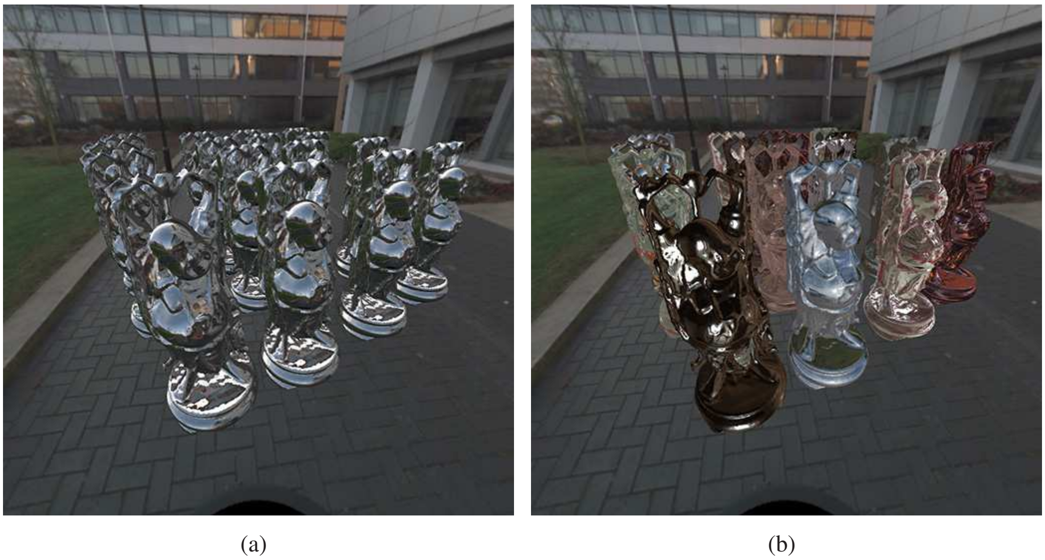


Figure 7.15: *A simple real-time application showing the Happy Buddha from the Stanford model repository with emphasis on texturing operations for timing: Top left the ground truth HDR texture was encoded using 32-bit floating point numbers for each channel (96 bpp). Top right RGBE encoding (32 bpp). Bottom left WETL (16 bpp), note that in this case the splitting axis between LDR and HDR image minimises RMSE error but appearance presents noticeable quantisation artefacts. Bottom right MWR (8 bpp).*



(a) (b)

Figure 7.16: *The cache test on a GeForceGO 7300: a) 16 Happy Buddhas rendered with a single HDR texture at 18.13 FPS. b) 16 Happy Buddhas rendered with 16 different HDR textures at 17.91 FPS.*

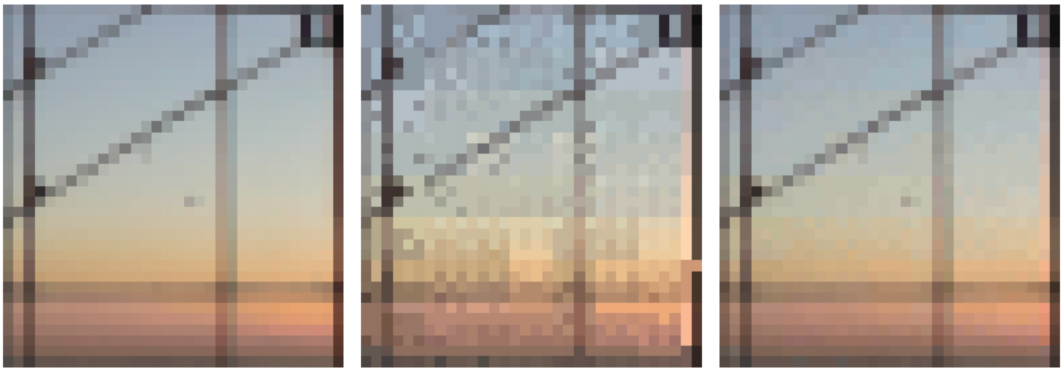
and 2 texture fetches.

Texture caching is an issue to take into account during the second texture fetch in Listing 1, because it depends on the first texture look-up. Since the 1D texture is quite small (512 bytes) it remains in the GPU cache. To verify this a simple scene was rendered with 16 Happy Buddhas with one single texture and the same scene with 16 different textures, see Figure 7.16. Performances were timed, and the results are 18.13 FPS for the single texture application, and 17.91 FPS for the application using 16 different textures. So the difference in terms of FPS is only 0.41 FPS, which is quite low and shows that the second fetch does not harm caching.

7.5 Summary

In this chapter, a compression framework for HDR textures was presented. The framework is based on tone mapping for the encoding stage and inverse tone mapping for the decoding stage. S3TC was employed for compressing LDR textures resulting in a compression rate of 8 bpp for high quality HDR textures and 4 bpp for acceptable HDR ones. Moreover, it can achieve real-time decoding performances for current graphics hardware. Although S3TC is the key for real-time applications, it is the main limit to the quality. Hence, MWOR and MWR methods are affected by the classical inherited problems with this compression scheme, see Figure 7.17.

MWR and MWOR have been designed for natural HDR textures, and tested with a maximum contrast ratio of $1.5 \times 10^8 : 1$ (in Image 11 from Figure 7.11). Images with contrast ratio over $10^9 : 1$ are a difficult scenario in which quantisation artefacts would be noticeable.



(a) (b) (c)

Figure 7.17: *A close-up of the central window in Image 12 in Figure 7.11. Note MWR and MWOR method does unfortunately inherit colour mixing artefacts from S3TC at edges and in a colour gradient. a) Original HDR image. b) Compressed image without residuals, colour mixing decreases the quality of luminance. c) Compressed image with residuals.*

MWR and MWOR have a good quality compared to existing available solutions for real-time applications on current hardware, RGBE and WETL, and the state of the art that needs special hardware METL and RETL. Furthermore, they can be implemented efficiently on new and old GPUs. Moreover, the decoding shader is easy to integrate into existing real-time applications. Finally, MWR and MWOR use at least half the memory of WETL avoiding contouring artefacts.

Chapter 8

Conclusions and Future Work

This thesis has introduced a framework that expands LDR content for the generation of HDR images and videos. The application of the framework was demonstrated through IBL and the visualisation of HDR content on HDR monitors.

The proposed solution fills the gap between classic imaging and HDR imaging, allowing existing LDR content to be used in HDR imaging. Even if HDR imaging has become very popular outside the computer graphics community, there are no still HDR cameras and video-cameras for consumer users. A certain knowledge, that a general consumer is not expected to have, is needed to capture HDR images. Moreover, only capturing still images requires certain conditions such as static scenes, a tripod to avoid misalignments, and no variation in the lighting conditions. To meet all these conditions is very difficult in real life and it can take a lot of time. Therefore, the proposed solution for the inverse tone mapping problem results in a good compromise between HDR imaging and available camera technology. Hence, the user needs only to capture normal images, and apply the framework to them.

8.1 Contributions

The framework in this thesis is the first solution to the inverse tone mapping problem that can be used for both visualisation of images on HDR monitors and IBL. The proposed framework expands the range of an LDR image applying an inverted TMO. Over-exposed areas are reconstructed using an expand map, which is a smooth field generated from areas of high luminance values. These areas are recovered by sampling light sources in the image. The framework can process videos without flickering using a temporally coherent expand map that re-uses samples of previous and forward frames in a video. Moreover, an efficient real-time implementation can be achieved exploiting graphics hardware through point based rendering.

The presented framework was validated with other expansion algorithms that convert LDR content to HDR content. Two studies were shown. The first one was a psychophysical study based on pairwise comparison experiments. The main result showed that operators handling artefacts performed better than straightforward operators. Furthermore, IBL needs a non-linear behaviour in order to reproduce a good approximation of the original environment maps. Finally, the proposed framework performed better than others for both the visualisation of images and IBL. The second study was based on computational metrics. In the first part the quality performances were analysed. The study partially confirmed the results of the psychophysical study with few differences. For example, Wang et al.'s operator [207] performed worse than before, because hallucinated areas were spotted by the metric. The second part of the study analysed if the framework introduces flickering in comparisons with Rempel's operator [171], the only other available operator for videos at the time of the study. This study showed that the framework is less prone to flickering especially for turbulent scenes where motion is very fast and dynamic, such as a fire, smoke, etc.

A compression scheme for textures can be derived from the framework, using tone mapping for encoding an HDR texture, and inverse tone mapping during the decoding stage. The compression scheme can achieve a 8 bpp compression rate, with a low error from the reference. The decompression scheme can be implemented on graphics hardware in real-time using few instructions and it can be integrated with minimal effort onto current applications. Moreover, an analysis of the error was carried out showing that tone mapped textures can be up-sampled before expansion, which in turn speeds-up the decoding time. Note that the application of inverse tone mapping to compression completes the set of LDR images that can be inverse tone mapped.

8.2 Limitations

The main limitations of the presented solution to inverse tone mapping depends on the size of over-exposed areas in the image or video. The quality is inversely proportional to the area of over-exposed regions. This is quite straightforward to realise, because a large over-exposed area implies more information to reconstruct than a smaller one. Reconstruction, not only of the range, can be reasonable if the area is small however it produces a smooth gradient which is not ideal for an image, see Figure 8.1.


(a) (b)

Figure 8.1: *An example of the limits of the proposed algorithm: a) The original LDR image. Note that presents large over-exposed areas. b) The expanded image in a) using the algorithm from Section 5.1 at f-stop* -3. *The reconstruction produced a smooth gray gradient pattern, because there is not enough information in these areas.*

For compression, the proposed encoding and decoding scheme cannot deal with large dynamic range, for example more than $10^8:1$, because quantisation artefacts will appear in the form of banding. This is because only 8 bits are used for the luminance channel and they cannot be enough for encoding a texture with that range of values. For dynamic ranges higher than that amount the technique could work by increasing the number of pixel bits and using a separated luminance colour space.

8.3 Future Work

The proposed framework for the expansion of LDR images and videos showed how to reconstruct missing content. However, a series of future improvements can be introduced to tackle the current limitations:

- **Exploitation of varying exposure in videos**. In a video sequence exposure changes in between frames, showing details in over-exposed or under-exposed areas which become well-exposed. For future work the idea is to project well-exposed areas in a previous or forward frame into the current one where this area is over-exposed or under-exposed.
- **Panorama Extraction in videos**. A video can be encoded extracting a panorama and moving objects, for example this fact is exploited in MPEG-7 to improve compression rate [29]. In the case of inverse tone mapping this encoding can be exploited to reduce over-exposed and under-exposed areas. A panorama from a video is created by blending different frames which usually have different exposures. The blending is typically implemented as an average. However, if a HDR capturing method is employed, as in Debevec and Malik's algorithm [47], an HDR panorama can be produced. This panorama would not probably capture the full range of the scene, but it could reduce under-exposed and over-exposed areas in the image. The reduction of these areas can help inverse tone mapping to improve the quality during the reconstruction of the missing content.

Moreover, a more in depth study on expanded videos would be an important direction for future work to asses the inverse tone mapping algorithm for videos. However, psychophysical studies on videos with complex stimuli, such as a shot from a movie, need to be run taking into account that participants' memory is limited. Therefore, a preliminary study for choosing the correct evaluation methodology has to be done.

Regarding the compression scheme, modifications on the blocks of the implementation can be carried out for improving the quality of compressed textures. The main bottleneck is S3TC, but currently there is no other LDR codec for texture with hardware support. A decoder for a different scheme can be implemented on graphics hardware using shaders. However this penalises decoding performances and makes the decoding shader more complex. An interesting direction of research would be to investigate the application of the algorithm to the compression of the frame buffer of a 3D application. For example, the standard frame buffer for a modern 3D game is full high definition resolution, $1,920 \times 1,080$, which means 24 Mb for a full HDR frame buffer. The compression scheme can be adapted for real-time compression. For example the optimisation step could be removed using not optimal TMO parameters, and a real-time S3TC encoder could be employed for compressing the tone mapped image such as the one proposed by van Waveren [199].

8.4 Final Remarks

HDR technology has not yet reached its full maturity. Capturing HDR videos is still an open issue. Moreover, capturing HDR still images can be a long and tedious task when using classic LDR technology, or very expensive with HDR cameras. The work of this thesis proposed a solution for the reconstruction of HDR content from LDR content. This simplifies the capturing of HDR content for consumer users. Moreover, all LDR videos and images captured previously can be utilised for HDR media, or used for HDR processing and re-lighting real and synthetic objects.

Appendix A

The Bilateral Filter

The bilateral filter is a non-linear filter proposed by Tomasi and Manduchi [191] which keeps strong edges while it smoothes inside values. The filter for an image I is defined as:

$$I'(\mathbf{x}) = B(I, f, g) = \frac{1}{k(\mathbf{x})} \sum_{\mathbf{y}\in\Omega} I(\mathbf{y}) f(\|\mathbf{x}-\mathbf{y}\|) g(\|I(\mathbf{y}) - I(\mathbf{x})\|) \tag{A.1}$$

$$k(\mathbf{x}) = \sum_{\mathbf{y}\in\Omega} f(\|\mathbf{x}-\mathbf{y}\|) g(\|I(\mathbf{y}) - I(\mathbf{x})\|) \tag{A.2}$$

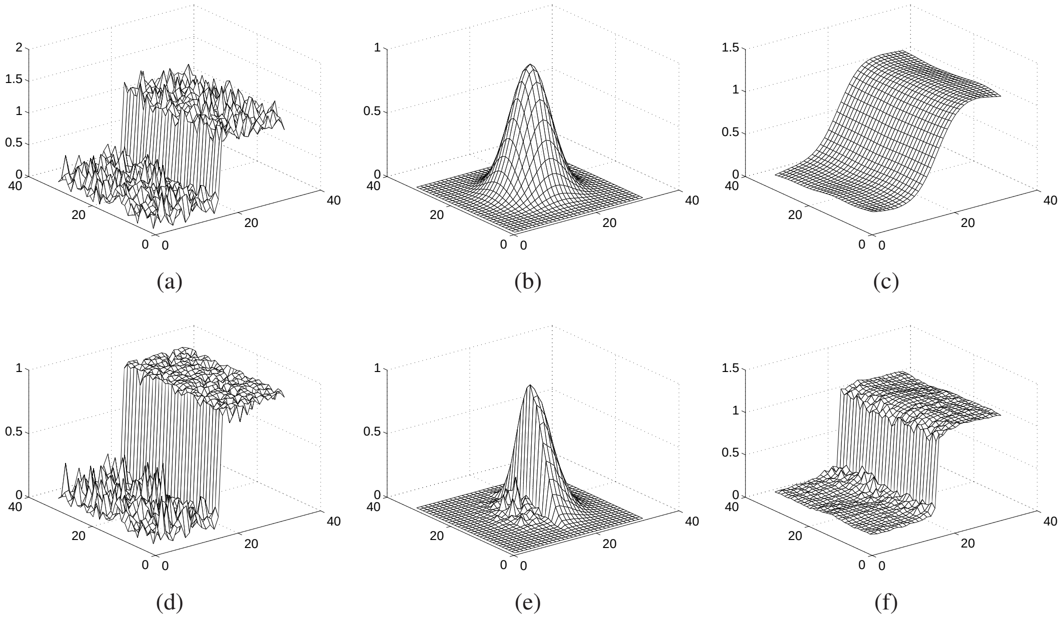
where I' is the filtered image, f is the smoothing function for the range term, and g is the smoothing function for the intensity term , $k(\mathbf{x})$ is the normalisation term. In the standard notation, while parameters for f are indicated with $_s$, the ones for g are indicated with $_r$. An example of the application of the filter is shown in Figure A.1.

Moreover, the bilateral filtering can be used for transferring edges from a source image to a target image modifying Equation A.2 as:

$$I'(\mathbf{x}) = B(I, J, f, g) = \frac{1}{k(\mathbf{x})} \sum_{\mathbf{y}\in\Omega} I(\mathbf{y}) f(\|\mathbf{x}-\mathbf{y}\|) g(\|J(\mathbf{y}) - J(\mathbf{x})\|) \tag{A.3}$$

$$k(\mathbf{x}) = \sum_{\mathbf{y}\in\Omega} f(\|\mathbf{x}-\mathbf{y}\|) g(\|J(\mathbf{y}) - J(\mathbf{x})\|) \tag{A.4}$$

where J is the image containing edges to be transferred to I. This version is called joint/cross bilateral filtering [160, 98]. When this version is used for up-sampling, it is called joint bilateral

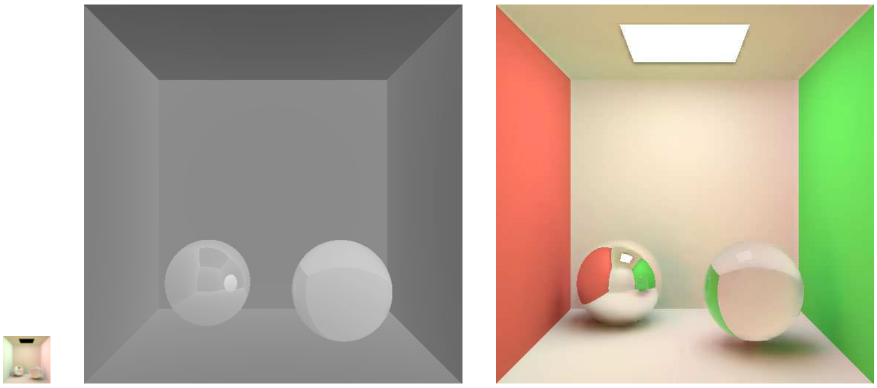


(a) (b) (c)

(d) (e) (f)

Figure A.1: *An example of bilateral filter: a) An edge corrupted by additive Gaussian noise with* $\sigma = 0.25$*. b) A spatial Gaussian kernel with* $\sigma_s = 4$ *evaluated at the central pixel. c) Image in a) filtered with the Gaussian kernel in b), note that noise is removed but edges are smoothed causing blur effect. d) An intensity Gaussian kernel with* $\sigma_r = 0.5$ *evaluated at the central pixel. e) A spatial-intensity Gaussian kernel obtained by multiplying kernel in b) and d), called bilateral kernel. f) Image in a) filtered using the bilateral kernel, note that while edges are preserved, noise is removed.*

up-sampling [98]. This is a general technique to speed-up computation of various trasks such as stereo matching, tone mapping, global illumination, etc. The task is computed in a small scale and then it is up-sampled using the starting full resolution image or other features (for example normal or depth values in the case of global illumination), see Figure A.2.



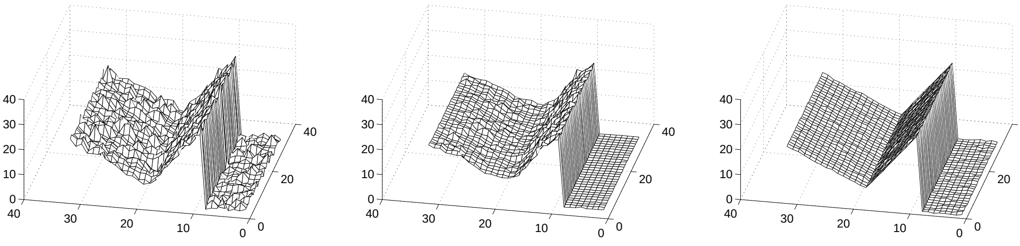
(a) (b) (c)

Figure A.2: *An example of joint bilateral up-sampling for rendering: a) The starting low resolution image representing indirect lighting. b) A depth map used as edge map. c) The up-sampled image in a) transferring edges of b) with direct lighting added.*

The main disadvantages of the bilateral filter are three. The first is that it smoothes across sharp changes in gradients blunting or blurring ramp edges and valley or ridge like feature. The second is that high gradient or high curvature regions are poorly smoothed because most nearby values are outliers that miss the filter window. The third is that it may include disjoint domains or high gradients regions. To solve all these problems the bilateral filter was extended by Choudhury and Tumblin [32]. The new filter, called trilateral filter, tilts and skews the filter window using bilateral filtered image gradients ∇I_θ. This is achieved by using as input for g the closeness of the centre value $I(\mathbf{x})$ to a plane istead of the Eucledian distance of coordinates. This plane is defined as:

$$P(\mathbf{x},\mathbf{y}) = I(\mathbf{x}) + \nabla I_\theta \cdot \mathbf{y} \tag{A.5}$$

where $\mathbf{x}$ are the coordinates of the point to filter, $\mathbf{y}$ are the coordinates of a sample in the window. An example of the trialteral filter is shown in Figure A.3. The only disadvantage of this filter is the high computational cost, because two bilateral filters are needed to be calculated, one for gradients and another for the filtering image values.



(a) (b) (c)

Figure A.3: *A comparison between bilateral and trilateral filter: a) The input noisy signal. b) The signal in a) smoothed using bilateral filter. c) The signal in a) smoothed using trilateral filter. Note that ramps and ridges are kept instead of being smoothed as it happened in b).*

Appendix B

S3TC: The de facto standard for LDR texture compression

S3TC is a block truncation coding scheme designed at S3 Graphics Ltd by Iourcha et al. [87] and inspired by Knittel et al.'s work [96]. The method has become the de facto standard for compression LDR texture in real-time applications, because S3TC and its variants are widely supported by graphics vendors. S3TC is the base of the proposed compression scheme in Chapter 7. Therefore, in this section, S3TC compression scheme will be presented.

S3TC is sometimes called DXTC, and it presents five modes where the difference is mainly in the coding of the alpha channel. The more popular variants are Mode 1 (DXT1) and Mode 5 (DXT5). The first step of the encoding for all modes is to convert an image from 24 bpp colour (8-bit for each channel) to a 16 bpp colour one. While 5 bits are assigned to red and blue channels, 6 bits are allocated for the green one. Then, 4×4 blocks of the texture are processed. For each block, the maximum, c_0, and the minimum, c_1, colours are calculated as values that most closely resemble the uncompressed block. Subsequently, C_0 and C_1 are stored. At this point, Mode 1 stores a 2-bit value, *ind*, for each pixel, which gives the nearest colour to the original one in the interpolation between c_0 and c_1. For a complete allocation of bits in Mode 1 see Table B.1. Mode 1 allows to have an 1 bit alpha channel, but precision for interpolation of colours is reduced for encoding alpha. Mode 5 is similar to Mode 1 without 1 bit alpha. Moreover, it can encode an 8-bit alpha channel in a similar way as for colours in Mode 1: the minimum, α_0, and maximum, α_1, alpha in the block, and a 3-bit value, ind_α, for each pixel used for the interpolation. The bit allocations for Mode 5 shares the same colour part of Mode 1 in Table B.1, plus there is a similar bit allocation for alpha values.

<table>
<tr><td>byte</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>0</td><td colspan="8">c_0</td></tr>
<tr><td>1</td><td colspan="8">c_0</td></tr>
<tr><td>2</td><td colspan="8">c_1</td></tr>
<tr><td>3</td><td colspan="8">c_1</td></tr>
<tr><td>4</td><td colspan="2">ind_0</td><td colspan="2">ind_1</td><td colspan="2">ind_2</td><td colspan="2">ind_3</td></tr>
<tr><td>5</td><td colspan="2">ind_4</td><td colspan="2">ind_5</td><td colspan="2">ind_6</td><td colspan="2">ind_7</td></tr>
<tr><td>6</td><td colspan="2">ind_8</td><td colspan="2">ind_9</td><td colspan="2">ind_{10}</td><td colspan="2">ind_{11}</td></tr>
<tr><td>7</td><td colspan="2">ind_{12}</td><td colspan="2">ind_{13}</td><td colspan="2">ind_{14}</td><td colspan="2">ind_{15}</td></tr>
</table>

<table>
<tr><td>byte</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>0</td><td colspan="8">α_0</td></tr>
<tr><td>1</td><td colspan="8">α_1</td></tr>
<tr><td>2</td><td colspan="3">$ind_{\alpha,0}$</td><td colspan="3">$ind_{\alpha,1}$</td><td colspan="2">$ind_{\alpha,2}$</td></tr>
<tr><td>3</td><td>$ind_{\alpha,2}$</td><td colspan="3">$ind_{\alpha,3}$</td><td colspan="3">$ind_{\alpha,4}$</td><td>$ind_{\alpha,5}$</td></tr>
<tr><td>4</td><td colspan="2">$ind_{\alpha,5}$</td><td colspan="3">$ind_{\alpha,6}$</td><td colspan="3">$ind_{\alpha,7}$</td></tr>
<tr><td>5</td><td colspan="3">$ind_{\alpha,8}$</td><td colspan="3">$ind_{\alpha,9}$</td><td colspan="2">$ind_{\alpha,10}$</td></tr>
<tr><td>6</td><td>$ind_{\alpha,10}$</td><td colspan="3">$ind_{\alpha,11}$</td><td colspan="3">$ind_{\alpha,12}$</td><td>$ind_{\alpha,13}$</td></tr>
<tr><td>7</td><td colspan="2">$ind_{\alpha,13}$</td><td colspan="3">$ind_{\alpha,14}$</td><td colspan="3">$ind_{\alpha,15}$</td></tr>
</table>

Table B.1: *The table shows the bit allocation for a* 4×4 *block in S3TC Mode 1.*

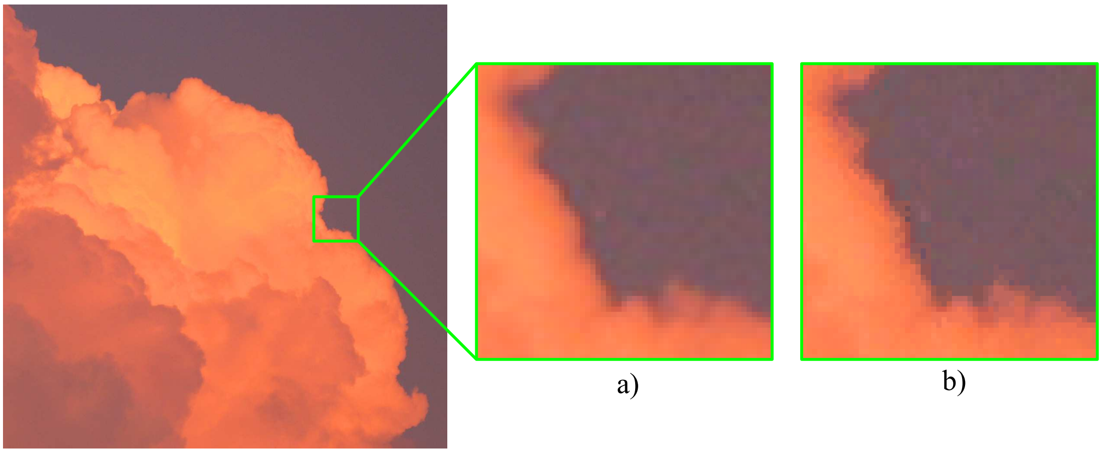

a) b)

Figure B.1: *An example of typical artifacts of S3TC for the Clouds image. On the left side the full resolution image. On the right side two zoom of the green box: a) A zoom of the original image. b) A zoom of the S3TC compressed image. Note that it is less smooth the one in a), with colour bleeding at edges and a general blocky appearance.*

When the colour of a pixel in a certain block needs to be decompressed, a linear interpolation between c_0 and c_1 is calculated:

$$Colour(\mathbf{y}) = \frac{1 - ind_{k(\mathbf{y})}}{3} c_0 + \frac{ind_{k(\mathbf{y})}}{3} c_1 \tag{B.1}$$

where $k(\mathbf{y})$ denotes the *ind* for pixel at position $\mathbf{y}$. Equation B.1 is applied for both Mode 1 and Mode 5. In the case of Mode 5, additionally to Equation B.1, alpha values are reconstructed using a similar interpolation scheme:

$$\alpha(\mathbf{y}) = \frac{1 - ind_{\alpha,k(\mathbf{y})}}{7} \alpha_0 + \frac{ind_{\alpha,k(\mathbf{y})}}{7} \alpha_1 \tag{B.2}$$

The main advantages of S3TC are the fixed rate compression and a single access to the memory for decompression. Therefore, a fast implementation on graphics hardware can be achieved. Moreover, it reaches good compression rates, 4 bpp for a 24 bpp image, and 8 bpp for a 32 bpp image with alpha channel. However, the algorithm can show artifacts when different colours are present in the same block, see Figure B.1.

Appendix C

An Overview on Graphics Processing Units Architectures

In the last few years graphics hardware have reached a high level of parallelism with peaks of up to a teraflop of power for desktop applications. This computational power allows to run extremely fast algorithms that are mapped to these architectures. The current graphics hardware is the main source of this power, GPUs are extremely parallelised SIMD hardware [145] with up to more than 200 cores per GPU against the 4 cores per CPU, see Figure C.1.

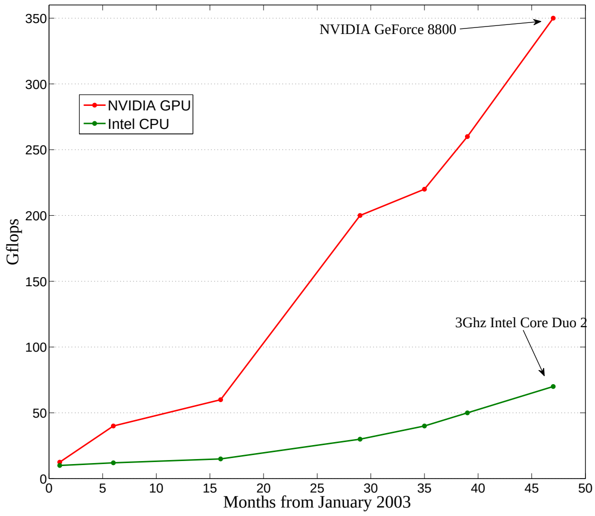


Figure C.1: *This graph shows the increasing computational power in GFlops of GPUs by NVIDIA and CPUs by Intel from January 2003. Note that GPUs have a higher rate than CPUs.*

Graphics processing units (GPUs) are optimised to render a stream of geometric primitives (point, lines and triangles), called *vertex buffer*, onto an array of pixels called *frame buffer*. The GPUs became fully programmable in the last few years for transforming and lighting vertices and pixels. The main purpose of the GPU is to process vertices and pixels. This processing follows the classic graphics pipeline, allowing in certain stages programmability via micro programs, see Figure C.2 for an example of GPU pipeline.

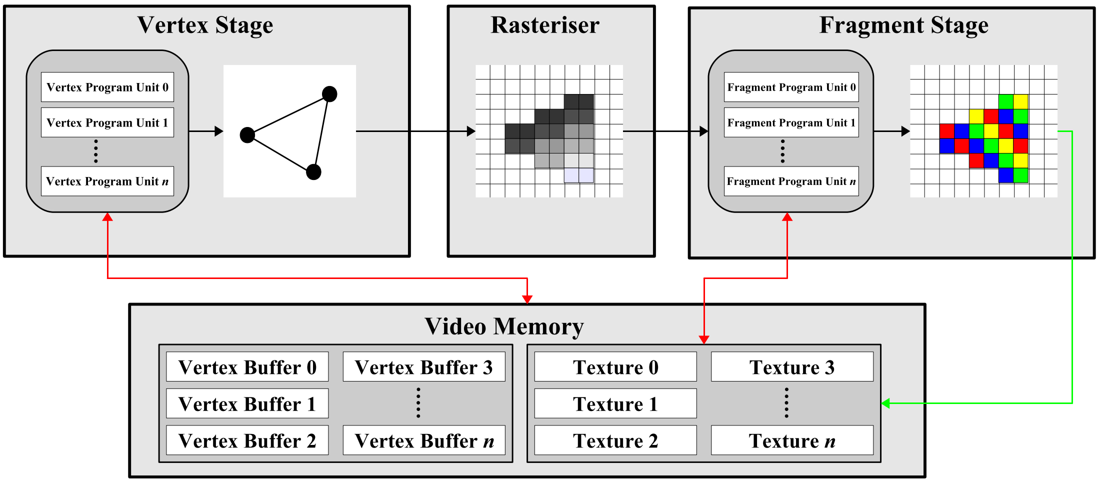


Figure C.2: *The GPU Pipeline: vertices which define a primitive (triangle, line, a point) are transformed using a function (implemented as a Vertex Program) in the Vertex Stage by Vertex Program Unit. Vertices are elaborated in parallel by many Vertex Program Units (Vertex Program Unit* 0, ..., *Vertex Program Unit* n*). Then primitives are discretised in fragments by the Rasteriser which passes these fragments to the Fragment Stage. A function (Fragment Program) is applied to each fragment in parallel using many Fragment Program Units (Fragment Program Unit* 0, ..., *Fragment Program Unit* n*). The result of the Fragment Stage is saved in a texture in the video memory.*

The first step in the pipeline is to process vertices. Each vertex of a vertex buffer is processed by a Vertex Program Unit (VPU) which is a programmable unit that executes a vertex program (VP). A VP is a set of instructions that specifies how a vertex will be processed by the VPU. The output of a VPU is not only a modified position for a vertex but it can also include new added properties to the vertex during the execution of the VP such as a colour, an address for fetching the memory during the next phase, etc. Subsequently, a unit called rasteriser, generates fragments of the primitives, in other words it discretises them into pixels and interpolates values between vertices using linear interpolation. In the last step of the pipeline fragments are processed by another programmable unit called Fragment Program Unit (FPU), which executes a fragment program (FP). Much the same as VP, the FP is a set of instructions which specifies how a fragment will be processed by the FPU. The output, which can be a single value or a vector of values, can be stored in the frame buffer for visualisation or in a texture for future processing. Note that a modern GPU presents only a single kind of unit, processing unit (PU), that can be allocated for executing VP or FP. In this current generation of GPUs the main instructions allowed in the VPs and FPs are: float point/integer operations (addition, subtraction, division, multiplication, square root, logarithm, etc.), random access to the video memory, assignment command, static and dynamic branching (a costly operation), and loop (with limited loop size for avoiding infinite loops).

VPs and FPs are usually written in a high-level programming language similar to C. These languages are called *shading languages* because they are designed for generating images. The most common shading languages are: C for Graphics (Cg) [128], the OpenGL Shading Language (GLSL) [94] and High Level Shading Language (HLSL) [22]. These languages provide an abstraction for a very close level to the hardware, hence they directly manage vertices, textures, fragments, etc. Recently, a new API called CUDA was introduced [145]. This API is designed for GP-GPU and it works directly with the hardware without interfacing with a Graphics API. The main drawback of CUDA is that is not retro-compatible with previous GPUs, indeed it works only on NVIDIA hardware from GeForce series 8 architecture onwards.

The paradigm of GPU programming is called Kernel Programming [149], where a kernel is the inner loop of a function applied to all pixels/vertices. These are the following steps for running a kernel:

1. Vertices are passed to the GPU, in order to feed the vertex stage. A typical image processing invocation is a quadrilateral, parallel to the screen of the display, which covers a region of pixels that match precisely with the desired size of the output texture. This provides the extreme boundaries (a quadrilateral) of a temporary address space for allocating images.

2. The rasteriser generates a fragment for every pixel in the quadrilateral. The rasteriser fills the address space with all the addresses, called fragments.

3. Each fragment is processed by the FPU. The fragment program can arbitrarily read from textures in the video memory, but can only write to memory corresponding to the location of the fragment in the frame buffer determined by the rasteriser. The domain of the

computation is specified for each input texture by specifying texture coordinates at each of the input vertices of the quadrilateral.

4. The output of the fragment program is a value, that is stored in a texture.

An algorithm needs to reiterate this process as many times as required by the computation. This is called *multi-pass*.

Appendix D

Tables of Psychophysical Experiments: Experiment 1 Overall

Scene 1						
	A	B	M	W	R	Score a_i
A	-	7	20	9	9	45
B	17	-	20	21	15	73
M	4	4	-	6	5	19
W	15	3	18	-	14	50
R	15	9	19	10	-	53

Scene 2						
	A	B	M	W	R	Score a_i
A	-	15	13	19	19	66
B	9	-	16	10	14	49
M	11	8	-	12	9	40
W	5	14	12	-	11	42
R	5	10	15	13	-	43

Scene 3						
	A	B	M	W	R	Score a_i
A	-	1	3	1	4	9
B	23	-	21	8	14	66
M	21	3	-	0	3	27
W	23	16	24	-	14	77
R	20	10	21	10	-	61

Scene 4						
	A	B	M	W	R	Score a_i
A	-	0	1	0	1	2
B	24	-	24	15	16	79
M	23	0	-	1	1	25
W	24	9	23	-	16	72
R	23	8	23	8	-	62

Scene 5						
	A	B	M	W	R	Score a_i
A	-	10	19	14	18	61
B	14	-	20	15	17	66
M	5	4	-	5	7	21
W	10	9	19	-	11	49
R	6	7	17	13	-	43

Scene 6						
	A	B	M	W	R	Score a_i
A	-	0	0	4	4	8
B	24	-	19	21	19	83
M	24	5	-	10	11	50
W	20	3	14	-	13	50
R	20	5	13	11	-	49

Scene 7						
	A	B	M	W	R	Score a_i
A	-	2	0	8	6	16
B	22	-	8	16	15	61
M	24	16	-	19	18	77
W	16	8	5	-	13	42
R	18	9	6	11	-	44

Scene 8						
	A	B	M	W	R	Score a_i
A	-	7	16	10	9	42
B	17	-	15	10	12	54
M	8	9	-	6	13	36
W	14	14	18	-	6	52
R	15	12	11	18	-	56

Table D.1: *Preference tables for each Scene used for the Overall experiment.*

	Scene 1	Scene 2	Scene 3	Scene 4	Scene 5	Scene 6	Scene 7	Scene 8
Subject 1	0.600	0.800	1.000	1.000	0.600	1.000	1.000	0.200
Subject 2	0.600	0.800	1.000	1.000	1.000	1.000	0.600	0.400
Subject 3	1.000	0.400	0.800	1.000	1.000	0.600	0.600	0.400
Subject 4	0.200	0.200	0.200	0.800	1.000	0.200	0.400	0.200
Subject 5	0.800	0.600	0.800	0.800	0.600	0.600	0.200	0.400
Subject 6	0.600	0.800	0.400	1.000	1.000	1.000	1.000	0.600
Subject 7	0.800	0.400	1.000	1.000	1.000	1.000	0.200	0.600
Subject 8	1.000	0.800	1.000	0.600	1.000	1.000	0.600	0.600
Subject 9	0.200	0.600	1.000	1.000	1.000	0.800	0.600	0.600
Subject 10	1.000	0.200	1.000	0.600	0.800	0.800	0.600	0.600
Subject 11	0.400	1.000	1.000	1.000	1.000	1.000	0.600	0.600
Subject 12	1.000	0.200	0.400	1.000	1.000	1.000	1.000	0.400
Subject 13	1.000	0.200	0.600	1.000	1.000	1.000	0.600	0.800
Subject 14	0.200	0.600	1.000	1.000	1.000	1.000	1.000	0.800
Subject 15	1.000	0.400	0.800	1.000	0.800	0.600	0.800	0.600
Subject 16	0.600	0.200	0.800	1.000	1.000	1.000	0.200	0.400
Subject 17	1.000	0.400	0.400	1.000	1.000	0.600	0.800	0.400
Subject 18	0.600	0.400	0.400	1.000	0.600	1.000	0.600	0.200
Subject 19	1.000	0.400	1.000	0.800	0.800	1.000	0.600	0.400
Subject 20	0.600	0.600	0.800	0.800	0.600	0.800	0.600	0.200
Subject 21	0.600	0.200	1.000	0.800	0.600	1.000	1.000	1.000
Subject 22	0.400	0.200	1.000	0.800	0.400	1.000	1.000	0.200
Subject 23	1.000	1.000	1.000	1.000	0.800	1.000	1.000	0.600
Subject 24	0.600	0.800	0.600	1.000	1.000	0.800	1.000	0.800

Table D.2: *Consistency tables used for the Overall Experiment 1.*

Appendix E

Tables of Psychophysical Experiments: Experiment 1 Bright Areas

Scene 1						
	A	B	M	W	R	Score a_i
A	-	6	24	5	9	44
B	18	-	24	24	19	85
M	0	0	-	1	3	4
W	19	0	23	-	21	63
R	15	5	21	3	-	44

Scene 2						
	A	B	M	W	R	Score a_i
A	-	12	19	15	19	65
B	12	-	23	14	18	67
M	5	1	-	2	0	8
W	9	10	22	-	12	53
R	5	6	24	12	-	47

Scene 3						
	A	B	M	W	R	Score a_i
A	-	3	5	2	9	19
B	21	-	20	12	18	71
M	19	4	-	3	9	35
W	22	12	21	-	13	68
R	15	6	15	11	-	47

Scene 4						
	A	B	M	W	R	Score a_i
A	-	3	1	1	9	14
B	21	-	14	14	17	66
M	23	10	-	9	10	52
W	23	10	15	-	18	66
R	15	7	14	6	-	42

Scene 5						
	A	B	M	W	R	Score a_i
A	-	5	24	11	16	56
B	19	-	23	12	13	67
M	0	1	-	0	2	3
W	13	12	24	-	12	61
R	8	11	22	12	-	53

Scene 6						
	A	B	M	W	R	Score a_i
A	-	4	1	15	15	35
B	20	-	5	23	20	68
M	23	19	-	23	19	84
W	9	1	1	-	14	25
R	9	4	5	10	-	28

Scene 7						
	A	B	M	W	R	Score a_i
A	-	8	3	11	16	38
B	16	-	7	19	21	63
M	21	17	-	19	14	71
W	13	5	5	-	14	37
R	8	3	10	10	-	31

Scene 8						
	A	B	M	W	R	Score a_i
A	-	10	19	9	11	49
B	14	-	24	14	15	67
M	5	0	-	4	3	12
W	15	10	20	-	9	54
R	13	9	21	15	-	58

Table E.1: *Preference tables for each Scene used for the Bright Areas experiment.*

	Scene 1	Scene 2	Scene 3	Scene 4	Scene 5	Scene 6	Scene 7	Scene 8
Subject 1	1.000	1.000	0.400	1.000	1.000	0.400	0.600	0.600
Subject 2	1.000	1.000	0.600	1.000	1.000	1.000	1.000	0.200
Subject 3	0.400	0.600	1.000	1.000	1.000	0.800	0.600	0.600
Subject 4	0.200	1.000	0.200	0.600	0.800	0.200	0.600	0.800
Subject 5	0.600	0.800	1.000	0.800	1.000	1.000	0.400	0.800
Subject 6	0.800	1.000	0.400	0.800	0.600	1.000	0.600	0.200
Subject 7	1.000	0.200	1.000	1.000	1.000	1.000	0.800	0.200
Subject 8	1.000	1.000	1.000	1.000	0.600	0.800	0.600	0.600
Subject 9	0.800	0.600	1.000	0.400	0.600	0.800	0.600	0.600
Subject 10	1.000	0.600	0.600	0.800	0.800	0.800	0.400	0.800
Subject 11	0.800	0.600	1.000	1.000	1.000	1.000	0.600	0.800
Subject 12	1.000	0.400	0.800	0.400	1.000	0.800	0.600	1.000
Subject 13	1.000	0.600	1.000	0.400	1.000	1.000	0.400	0.800
Subject 14	1.000	0.600	0.800	0.800	0.400	0.600	1.000	0.800
Subject 15	0.400	0.400	1.000	1.000	1.000	0.800	0.400	0.600
Subject 16	0.200	0.000	0.800	0.200	0.600	1.000	0.200	0.200
Subject 17	1.000	1.000	0.400	1.000	1.000	0.800	0.600	1.000
Subject 18	1.000	1.000	0.200	1.000	0.400	1.000	0.800	0.200
Subject 19	1.000	0.600	0.800	1.000	1.000	1.000	0.400	0.400
Subject 20	1.000	0.800	0.400	0.600	1.000	0.600	0.600	0.600
Subject 21	1.000	1.000	0.600	0.600	0.800	1.000	0.600	1.000
Subject 22	0.600	0.800	1.000	1.000	1.000	1.000	1.000	0.800
Subject 23	1.000	0.600	0.600	1.000	1.000	1.000	0.600	1.000
Subject 24	0.600	1.000	0.400	0.200	1.000	0.600	1.000	0.800

Table E.2: *Consistency tables used for the Bright Areas Experiment 1.*

Appendix F

Tables of Psychophysical Experiments: Experiment 1 Dark Areas

Scene 1						
	A	B	M	W	R	Score a_i
A	-	10	2	16	9	37
B	14	-	4	11	10	39
M	22	20	-	20	20	82
W	8	13	4	-	6	31
R	15	14	4	18	-	51

Scene 2						
	A	B	M	W	R	Score a_i
A	-	24	2	20	18	64
B	0	-	0	13	11	24
M	22	24	-	23	21	90
W	4	11	1	-	11	27
R	6	13	3	13	-	35

Scene 3						
	A	B	M	W	R	Score a_i
A	-	1	2	0	2	5
B	23	-	24	12	10	69
M	22	0	-	0	0	22
W	24	12	24	-	14	74
R	22	14	24	10	-	70

Scene 4						
	A	B	M	W	R	Score a_i
A	-	0	4	0	2	6
B	24	-	24	13	14	75
M	20	0	-	0	0	20
W	24	11	24	-	12	71
R	22	10	24	12	-	68

Scene 5						
	A	B	M	W	R	Score a_i
A	-	18	5	17	16	56
B	6	-	1	13	15	35
M	19	23	-	20	18	80
W	7	11	4	-	14	36
R	8	9	6	10	-	33

Scene 6						
	A	B	M	W	R	Score a_i
A	-	2	2	7	3	14
B	22	-	18	18	17	75
M	22	6	-	8	9	45
W	17	6	16	-	15	54
R	21	7	15	9	-	52

Scene 7						
	A	B	M	W	R	Score a_i
A	-	5	3	5	7	20
B	19	-	9	16	11	55
M	21	15	-	20	19	75
W	19	8	4	-	14	45
R	17	13	5	10	-	45

Scene 8						
	A	B	M	W	R	Score a_i
A	-	10	15	13	5	43
B	14	-	5	13	7	39
M	9	19	-	16	19	63
W	11	11	8	-	14	44
R	19	17	5	10	-	51

Table F.1: *Preference tables for each Scene used for the Dark Areas experiment.*

	Scene 1	Scene 2	Scene 3	Scene 4	Scene 5	Scene 6	Scene 7	Scene 8
Subject 1	0.400	1.000	0.800	0.800	0.800	0.600	1.000	0.400
Subject 2	0.600	1.000	1.000	1.000	0.800	1.000	0.800	0.400
Subject 3	0.800	0.800	1.000	1.000	0.600	0.400	0.200	0.000
Subject 4	0.600	0.400	0.600	0.600	0.400	0.800	0.600	0.200
Subject 5	0.600	0.800	0.800	1.000	0.600	0.800	0.000	0.600
Subject 6	0.200	0.800	1.000	0.800	0.400	1.000	0.600	0.400
Subject 7	1.000	1.000	0.400	1.000	0.800	0.800	0.200	0.400
Subject 8	0.400	1.000	0.800	0.800	0.600	1.000	0.400	0.200
Subject 9	1.000	1.000	1.000	1.000	0.200	0.800	0.600	0.200
Subject 10	1.000	0.800	1.000	1.000	0.800	0.200	0.600	0.400
Subject 11	0.800	1.000	1.000	1.000	0.400	1.000	0.400	1.000
Subject 12	0.400	0.400	0.600	0.400	0.800	1.000	0.800	0.400
Subject 13	1.000	1.000	1.000	1.000	0.400	1.000	0.600	0.600
Subject 14	0.400	1.000	0.800	1.000	1.000	1.000	1.000	0.600
Subject 15	0.600	1.000	1.000	0.800	0.800	1.000	0.600	0.400
Subject 16	0.400	0.800	0.800	1.000	1.000	1.000	0.600	0.200
Subject 17	0.600	0.400	1.000	0.800	0.600	0.600	0.600	1.000
Subject 18	0.400	0.400	1.000	1.000	0.600	1.000	0.600	0.600
Subject 19	0.200	1.000	1.000	0.800	0.600	1.000	0.600	0.600
Subject 20	0.600	0.400	1.000	1.000	1.000	0.800	1.000	1.000
Subject 21	0.600	0.800	1.000	0.800	1.000	0.400	0.200	0.600
Subject 22	1.000	0.600	0.800	1.000	1.000	0.800	1.000	0.600
Subject 23	0.800	0.800	1.000	0.800	0.400	0.600	0.600	0.200
Subject 24	0.600	0.600	1.000	0.800	0.200	0.800	0.800	0.400

Table F.2: *Consistency tables used for the Dark Areas Experiment 1.*

Appendix G

Tables of Psychophysical Experiments: Experiment 2 Diffuse Material

Scene 1						
	A	B	M	W	R	Score a_i
A	-	14	11	12	9	46
B	10	-	19	14	14	57
M	13	5	-	17	8	43
W	12	10	7	-	5	34
R	15	10	16	19	-	60

Scene 2						
	A	B	M	W	R	Score a_i
A	-	1	2	1	3	7
B	23	-	19	15	10	67
M	22	5	-	11	9	47
W	23	9	13	-	9	54
R	21	14	15	15	-	65

Scene 3						
	A	B	M	W	R	Score a_i
A	-	3	8	4	7	22
B	21	-	13	10	11	55
M	16	11	-	11	10	48
W	20	14	13	-	13	60
R	17	13	14	11	-	55

Scene 4						
	A	B	M	W	R	Score a_i
A	-	7	12	2	9	30
B	17	-	17	15	9	58
M	12	7	-	7	9	35
W	22	9	17	-	15	63
R	15	15	15	9	-	54

Scene 5						
	A	B	M	W	R	Score a_i
A	-	14	12	13	18	57
B	10	-	10	17	20	57
M	12	14	-	9	21	56
W	11	7	15	-	19	52
R	6	4	3	5	-	18

Scene 6						
	A	B	M	W	R	Score a_i
A	-	9	15	15	17	56
B	15	-	15	16	15	61
M	9	9	-	13	20	51
W	9	8	11	-	14	42
R	7	9	4	10	-	30

Table G.1: *Preference tables for each Scene used for the Diffuse Material experiment.*

	Scene 1	Scene 2	Scene 3	Scene 4	Scene 5	Scene 6
Subject 1	1.000	1.000	1.000	0.400	0.600	0.800
Subject 2	0.200	0.800	1.000	1.000	0.600	0.600
Subject 3	0.600	0.800	1.000	0.400	0.600	1.000
Subject 4	0.600	0.200	0.600	1.000	0.400	0.400
Subject 5	0.400	0.600	0.200	0.200	0.400	1.000
Subject 6	1.000	0.600	0.800	0.800	0.200	1.000
Subject 7	0.400	0.600	1.000	0.600	0.800	1.000
Subject 8	0.200	0.800	1.000	1.000	0.800	1.000
Subject 9	0.800	0.800	0.600	0.800	1.000	1.000
Subject 10	0.200	0.600	0.400	0.200	0.600	0.800
Subject 11	1.000	0.800	0.800	0.600	1.000	1.000
Subject 12	0.600	0.600	0.400	0.200	0.800	0.800
Subject 13	0.600	0.600	0.600	0.200	0.400	0.400
Subject 14	0.800	1.000	0.400	0.600	0.200	0.200
Subject 15	0.600	0.400	0.400	0.800	0.600	0.600
Subject 16	1.000	1.000	0.600	0.800	0.800	0.200
Subject 17	0.400	0.800	0.600	0.400	1.000	0.800
Subject 18	0.600	0.600	0.400	0.600	1.000	1.000
Subject 19	1.000	0.800	0.200	0.600	0.400	0.400
Subject 20	1.000	0.600	0.600	1.000	0.200	1.000
Subject 21	0.800	0.800	1.000	0.400	1.000	0.600
Subject 22	0.800	0.800	1.000	0.800	0.400	0.600
Subject 23	0.800	1.000	0.800	0.600	0.200	0.200
Subject 24	0.400	0.800	0.400	0.400	0.600	0.400

Table G.2: *Consistency tables used for the Diffuse Material Experiment 2.*

Appendix H

Tables of Psychophysical Experiments: Experiment 2 Glossy Material

Scene 1						
	A	B	M	W	R	Score a_i
A	-	5	6	9	8	28
B	19	-	17	20	14	70
M	18	7	-	11	5	41
W	15	4	13	-	6	38
R	16	10	19	18	-	63

Scene 2						
	A	B	M	W	R	Score a_i
A	-	4	2	3	2	11
B	20	-	12	12	9	53
M	22	12	-	9	10	53
W	21	12	15	-	10	58
R	22	15	14	14	-	65

Scene 3						
	A	B	M	W	R	Score a_i
A	-	7	6	9	10	32
B	17	-	10	13	13	53
M	18	14	-	12	12	56
W	15	11	12	-	11	49
R	14	11	12	13	-	50

Scene 4						
	A	B	M	W	R	Score a_i
A	-	10	8	8	9	35
B	14	-	18	18	9	59
M	16	6	-	8	11	41
W	16	6	16	-	14	52
R	15	15	13	10	-	53

Scene 5						
	A	B	M	W	R	Score a_i
A	-	13	13	11	19	56
B	11	-	11	18	16	56
M	11	13	-	10	18	52
W	13	6	14	-	16	49
R	5	8	6	8	-	27

Scene 6						
	A	B	M	W	R	Score a_i
A	-	7	10	12	17	46
B	17	-	17	14	14	62
M	14	7	-	7	10	38
W	12	10	17	-	19	58
R	7	10	14	5	-	36

Table H.1: *Preference tables for each Scene used for the Glossy Material experiment.*

	Scene 1	Scene 2	Scene 3	Scene 4	Scene 5	Scene 6
Subject 1	1.000	1.000	1.000	1.000	0.600	0.400
Subject 2	0.800	1.000	0.600	0.600	1.000	0.200
Subject 3	0.800	0.800	1.000	0.600	0.600	1.000
Subject 4	0.600	0.800	0.600	0.600	0.600	0.200
Subject 5	0.400	0.600	0.200	0.600	0.400	1.000
Subject 6	1.000	0.600	0.600	0.800	0.600	0.800
Subject 7	0.800	0.600	0.800	0.600	0.400	0.200
Subject 8	0.400	0.400	0.800	0.600	0.200	0.600
Subject 9	0.800	1.000	0.600	1.000	1.000	0.600
Subject 10	0.600	1.000	0.400	0.400	0.200	1.000
Subject 11	1.000	0.800	0.400	0.600	0.600	1.000
Subject 12	1.000	1.000	0.400	0.400	0.600	0.400
Subject 13	0.600	1.000	0.200	0.800	1.000	0.800
Subject 14	1.000	0.600	1.000	0.200	0.200	0.800
Subject 15	0.600	0.400	0.400	0.800	1.000	0.200
Subject 16	1.000	1.000	0.400	1.000	0.400	0.600
Subject 17	0.400	1.000	0.600	0.600	0.600	1.000
Subject 18	1.000	0.400	0.200	0.200	0.800	0.400
Subject 19	0.400	0.600	0.200	0.800	1.000	0.400
Subject 20	0.800	0.600	0.800	0.600	0.600	0.800
Subject 21	0.800	1.000	0.200	0.800	1.000	1.000
Subject 22	0.600	0.200	0.800	0.400	0.600	0.600
Subject 23	0.400	0.600	0.400	0.200	0.200	0.800
Subject 24	0.800	0.800	0.400	0.400	0.600	0.400

Table H.2: *Consistency tables used for the Glossy Material Experiment 2.*

Appendix I

Tables of Psychophysical Experiments: Experiment 2 Mirror Material

Scene 1						
	A	B	M	W	R	Score a_i
A	-	5	8	8	8	29
B	19	-	20	20	13	72
M	16	4	-	12	6	38
W	16	4	12	-	6	38
R	16	11	18	18	-	63

Scene 2						
	A	B	M	W	R	Score a_i
A	-	5	2	4	3	14
B	19	-	15	19	15	68
M	22	9	-	10	10	51
W	20	5	14	-	6	45
R	21	9	14	18	-	62

Scene 3						
	A	B	M	W	R	Score a_i
A	-	11	5	10	9	35
B	13	-	8	18	10	49
M	19	16	-	17	14	66
W	14	6	7	-	15	42
R	15	14	10	9	-	48

Scene 4						
	A	B	M	W	R	Score a_i
A	-	10	7	12	16	45
B	14	-	8	16	14	52
M	17	16	-	12	18	63
W	12	8	12	-	9	41
R	8	10	6	15	-	39

Scene 5						
	A	B	M	W	R	Score a_i
A	-	3	18	1	13	35
B	21	-	22	11	17	71
M	6	2	-	2	13	23
W	23	13	22	-	21	79
R	11	7	11	3	-	32

Scene 6						
	A	B	M	W	R	Score a_i
A	-	3	16	3	15	37
B	21	-	17	14	17	69
M	8	7	-	4	15	34
W	21	10	20	-	19	70
R	9	7	9	5	-	30

Table I.1: *Preference tables for each Scene used for the Mirror Material experiment.*

	Scene 1	Scene 2	Scene 3	Scene 4	Scene 5	Scene 6
Subject 1	1.000	1.000	1.000	0.600	1.000	1.000
Subject 2	1.000	1.000	1.000	0.800	1.000	0.600
Subject 3	0.600	0.800	1.000	0.200	1.000	0.800
Subject 4	0.400	0.400	0.200	0.200	0.400	0.600
Subject 5	0.600	0.600	0.200	0.400	0.800	0.600
Subject 6	1.000	0.600	0.800	0.800	0.600	1.000
Subject 7	0.400	1.000	0.600	0.400	1.000	0.600
Subject 8	0.600	0.400	0.800	0.800	0.800	0.600
Subject 9	0.800	0.800	0.200	0.200	1.000	0.800
Subject 10	0.800	0.800	1.000	0.400	1.000	1.000
Subject 11	0.800	0.800	0.200	0.200	1.000	0.800
Subject 12	1.000	0.400	1.000	0.600	1.000	1.000
Subject 13	0.200	0.400	1.000	0.200	1.000	0.600
Subject 14	1.000	0.400	0.600	0.400	0.800	1.000
Subject 15	0.600	0.400	0.400	0.800	1.000	0.600
Subject 16	0.800	1.000	0.600	0.600	0.800	1.000
Subject 17	1.000	1.000	0.800	1.000	0.800	1.000
Subject 18	0.600	0.600	0.600	0.200	1.000	1.000
Subject 19	0.400	1.000	1.000	0.200	0.200	0.600
Subject 20	0.600	0.600	0.800	0.600	0.200	0.600
Subject 21	1.000	0.800	0.600	1.000	1.000	0.400
Subject 22	0.200	1.000	0.800	0.800	0.800	0.000
Subject 23	0.200	0.200	0.800	0.400	0.200	0.200
Subject 24	0.600	0.600	0.400	1.000	0.800	0.800

Table I.2: *Consistency tables used for the Mirror Material Experiment 2.*

Appendix J

Images of Psychophysical Experiment 1

Figure J.1: *Scene 1.*

0

Figure J.2: *Scene 2.*

Figure J.3: *Scene 3.*

Figure J.4: *Scene 4.*

Figure J.5: *Scene 5.*

Figure J.6: *Scene 6.*

Figure J.7: *Scene 7.*

Figure J.8: *Scene 8.*

Appendix K

Images of Psychophysical Experiment 2

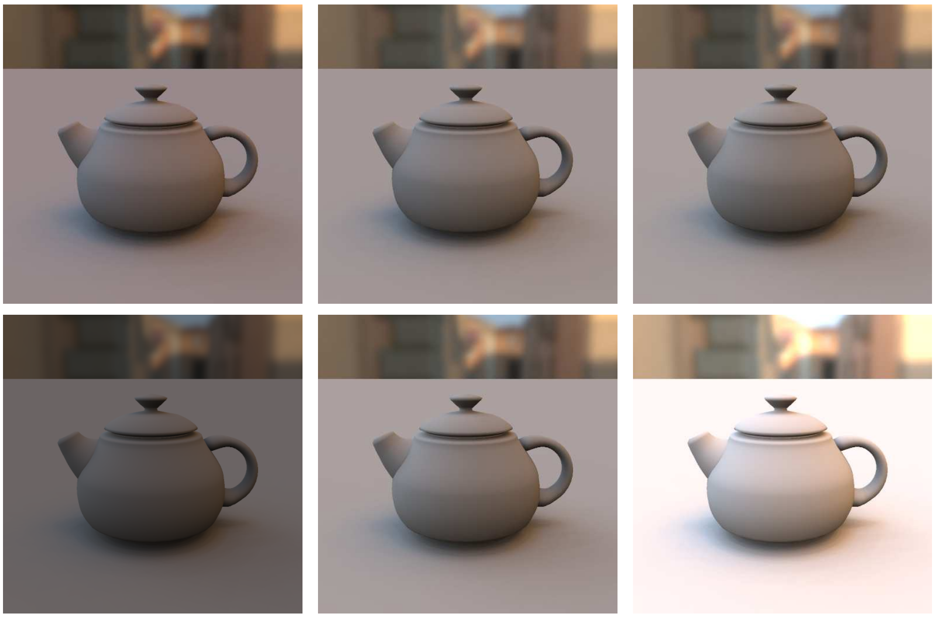


Figure K.1: *Images used in Experiment 2 using the Scene 1 environment map at exposure 0 for the diffuse material. In the first row from left to right: reference HDR environment map,* **R** *operator and* **B** *operator. In the second row from left to right:* **M** *operator,* **W** *operator and* **A** *operator.*

Figure K.2: *Images used in Experiment 2 using the Scene 1 environment map at exposure 0 for the glossy material. In the first row from left to right: reference HDR environment map,* **R** *operator and* **B** *operator. In the second row from left to right:* **M** *operator,* **W** *operator and* **A** *operator.*

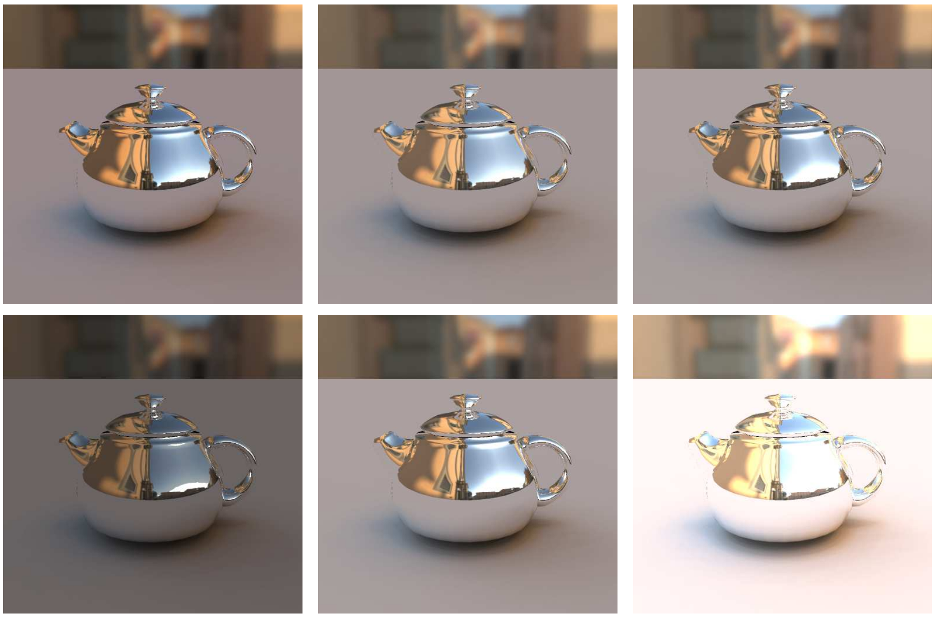


Figure K.3: *Images used in Experiment 2 using the Scene 1 environment map at exposure 0 for the pure specular material. In the first row from left to right: reference HDR environment map,* **R** *operator and* **B** *operator. In the second row from left to right:* **M** *operator,* **W** *operator and* **A** *operator.*

Figure K.4: *Images used in Experiment 2 using the Scene 2 environment map at exposure 0 for the diffuse material. In the first row from left to right: reference HDR environment map,* **R** *operator and* **B** *operator. In the second row from left to right:* **M** *operator,* **W** *operator and* **A** *operator.*

Figure K.5: *Images used in Experiment 2 using the Scene 2 environment map at exposure 0 for the glossy material. In the first row from left to right: reference HDR environment map,* **R** *operator and* **B** *operator. In the second row from left to right:* **M** *operator,* **W** *operator and* **A** *operator.*

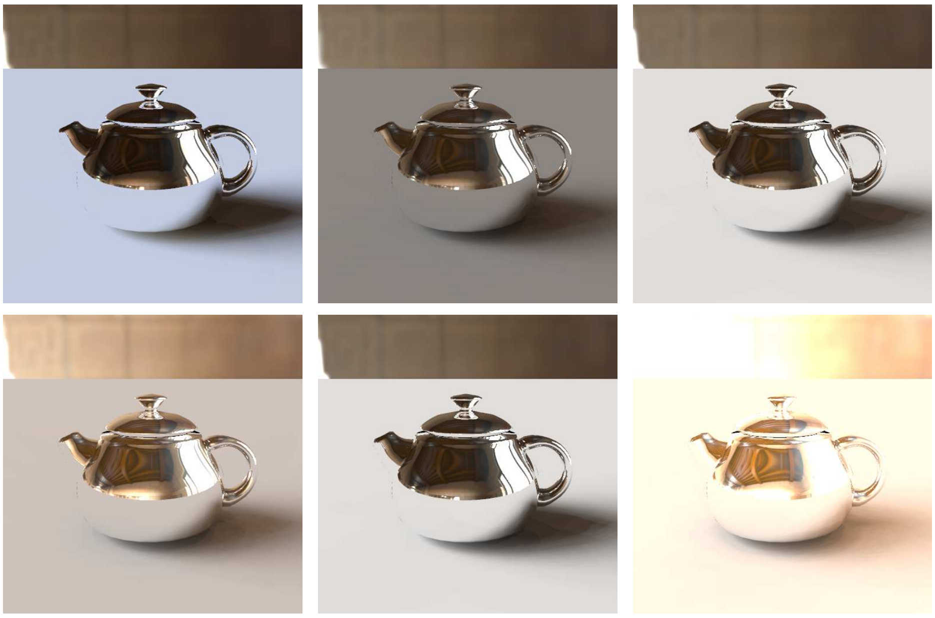


Figure K.6: *Images used in Experiment 2 using the Scene 2 environment map at exposure 0 for the pure specular material. In the first row from left to right: reference HDR environment map,* **R** *operator and* **B** *operator. In the second row from left to right:* **M** *operator,* **W** *operator and* **A** *operator.*

Figure K.7: *Images used in Experiment 2 using the Scene 3 environment map at exposure 0 for the diffuse material. In the first row from left to right: reference HDR environment map,* **R** *operator and* **B** *operator. In the second row from left to right:* **M** *operator,* **W** *operator and* **A** *operator.*

Figure K.8: *Images used in Experiment 2 using the Scene 3 environment map at exposure 0 for the glossy material. In the first row from left to right: reference HDR environment map,* **R** *operator and* **B** *operator. In the second row from left to right:* **M** *operator,* **W** *operator and* **A** *operator.*

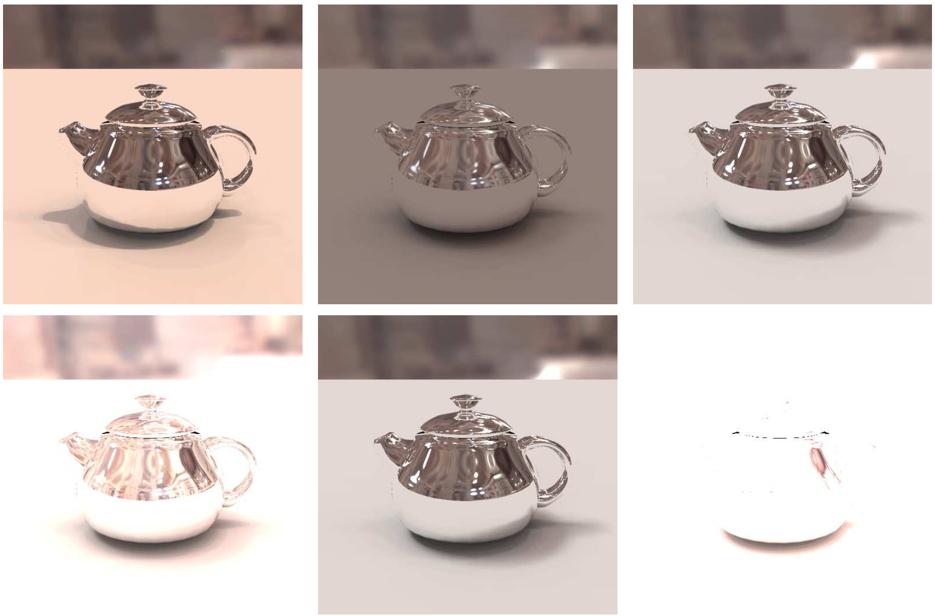


Figure K.9: *Images used in Experiment 2 using the Scene 3 environment map at exposure 0 for the pure specular material. In the first row from left to right: reference HDR environment map,* **R** *operator and* **B** *operator. In the second row from left to right:* **M** *operator,* **W** *operator and* **A** *operator.*

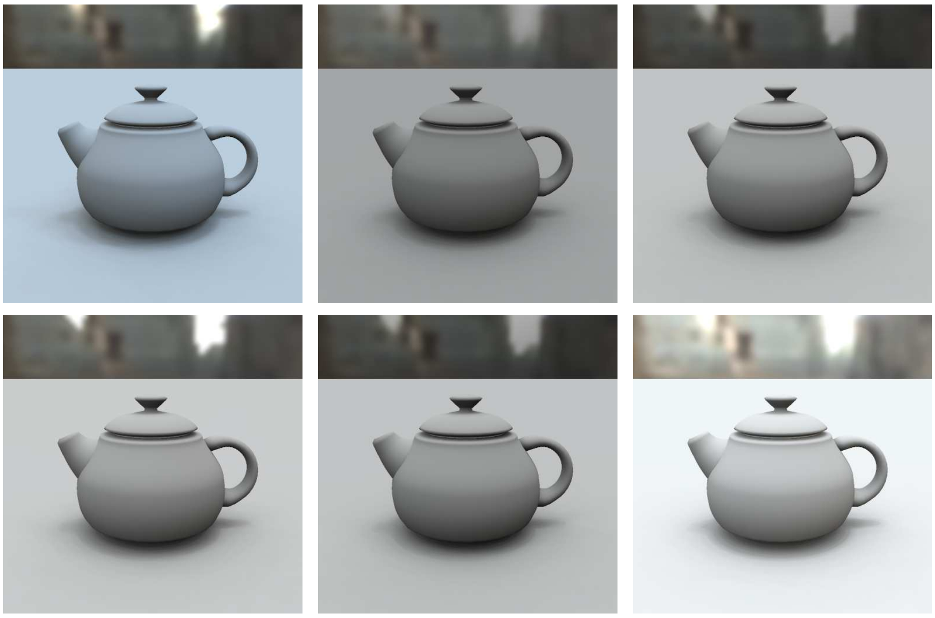


Figure K.10: *Images used in Experiment 2 using the Scene 3 environment map at exposure 0 for the diffuse material. In the first row from left to right: reference HDR environment map,* **R** *operator and* **B** *operator. In the second row from left to right:* **M** *operator,* **W** *operator and* **A** *operator.*

Figure K.11: *Images used in Experiment 2 using the Scene 4 environment map at exposure 0 for the glossy material. In the first row from left to right: reference HDR environment map,* **R** *operator and* **B** *operator. In the second row from left to right:* **M** *operator,* **W** *operator and* **A** *operator.*

Figure K.12: *Images used in Experiment 2 using the Scene 4 environment map at exposure 0 for the pure specular material. In the first row from left to right: reference HDR environment map,* **R** *operator and* **B** *operator. In the second row from left to right:* **M** *operator,* **W** *operator and* **A** *operator.*

Figure K.13: *Images used in Experiment 2 using the Scene 5 environment map at exposure 0 for the diffuse material. In the first row from left to right: reference HDR environment map,* **R** *operator and* **B** *operator. In the second row from left to right:* **M** *operator,* **W** *operator and* **A** *operator.*

Figure K.14: *Images used in Experiment 2 using the Scene 5 environment map at exposure 0 for the glossy material. In the first row from left to right: reference HDR environment map,* **R** *operator and* **B** *operator. In the second row from left to right:* **M** *operator,* **W** *operator and* **A** *operator.*

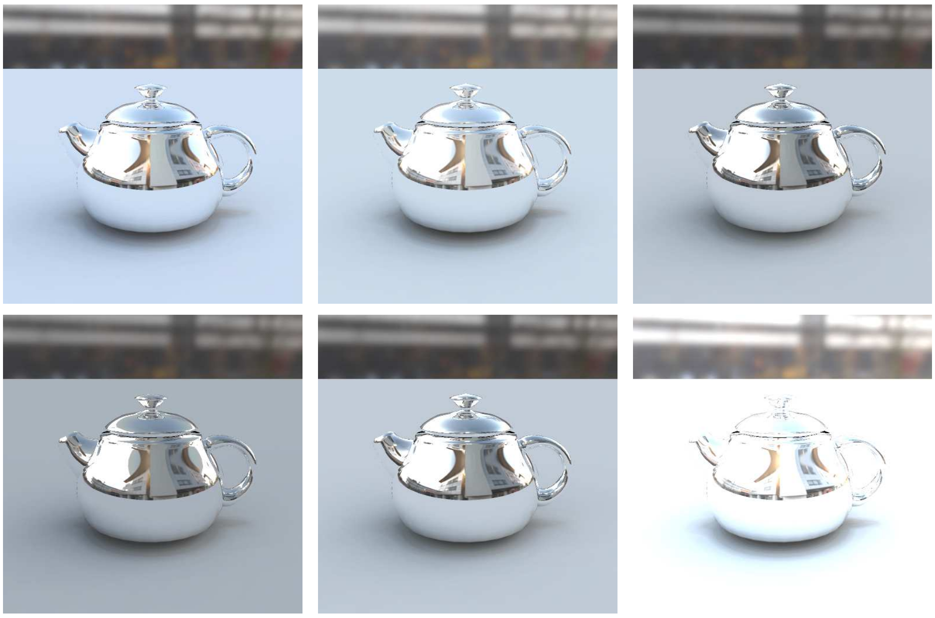


Figure K.15: *Images used in Experiment 2 using the Scene 5 environment map at exposure 0 for the pure specular material. In the first row from left to right: reference HDR environment map,* **R** *operator and* **B** *operator. In the second row from left to right:* **M** *operator,* **W** *operator and* **A** *operator.*

Figure K.16: *Images used in Experiment 2 using the Scene 6 environment map at exposure 0 for the diffuse material. In the first row from left to right: reference HDR environment map,* **R** *operator and* **B** *operator. In the second row from left to right:* **M** *operator,* **W** *operator and* **A** *operator.*

Figure K.17: *Images used in Experiment 2 using the Scene 6 environment map at exposure 0 for the glossy material. In the first row from left to right: reference HDR environment map,* **R** *operator and* **B** *operator. In the second row from left to right:* **M** *operator,* **W** *operator and* **A** *operator.*

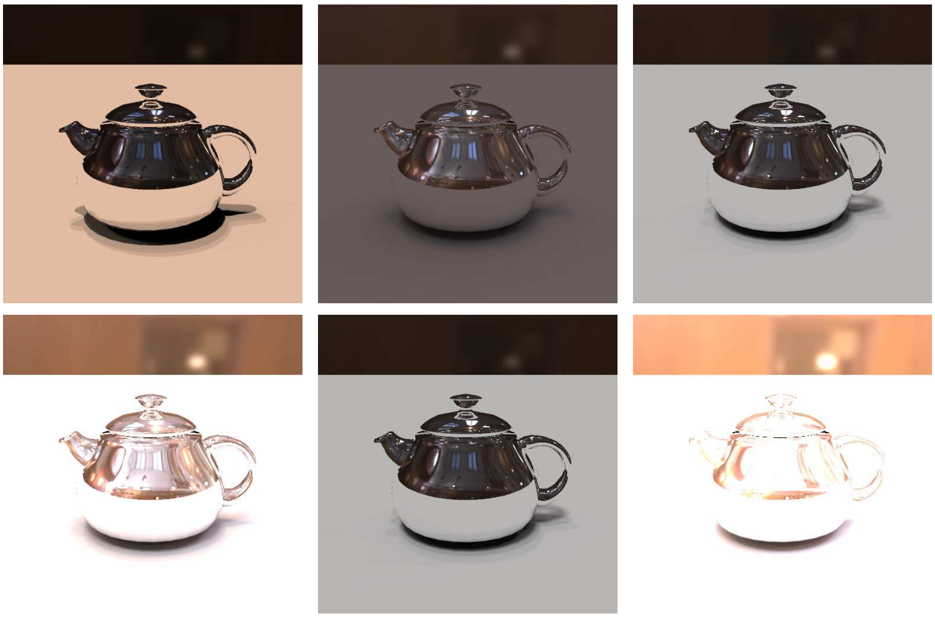


Figure K.18: *Images used in Experiment 2 using the Scene 6 environment map at exposure 0 for the pure specular material. In the first row from left to right: reference HDR environment map,* **R** *operator and* **B** *operator. In the second row from left to right:* **M** *operator,* **W** *operator and* **A** *operator.*

Appendix L

Abbreviations

- A: Akyüz et al.'s operator [7]
- B: Banterle et al.'s operator Section 5.1
- B_2: method in Section 5.2
- BRDF: Bi-directional Reectance Distribution Function
- BTC: Block Truncation Compression
- CAM: Colour Appearance Model
- CIE: Commission Internationale de lEclairage
- CRF: Camera Response Function
- DR-IIQA: Dynamic Range Independent Image Quality Assessment
- GPU: Graphics Processing Unit
- HDR: High Dynamic Range
- HVS: Human Visual System
- IBL: Image Based Lighting
- KMS: k-means sampling [97]
- iTMO: Inverse Tone Mapping Operator

- LDR: Low Dynamic Range
- M: Melyan et al.’s operator [133]
- MCS: median-cut sampling (MCS) [170]
- METL: Munkberg et al.s method [139]
- MWR: the proposed method with residuals
- MWOR: the proposed method without residuals
- RETL: Roimela et al.s method [174]
- PSNR: Peak Signal Noise Ratio
- PTS: Penrose tiling sampling [148]
- R: Rempel et al.’s operator [171]
- RI: Ratio Image
- RF: Reconstruction Function
- RMSE: Root Mean Squared Error
- SIS: structured importance sampling [4]
- TMO: Tone Mapping Operator
- VEP: Visual Equivalence Metric
- VDP: Visual Difference Predictor
- W: Wang et al.’s operator [207]
- WETL: Wang et al.s method [206]

Bibliography

[1] A. Adams. *The Print. The Ansel Adams Photography series.* Little, Brown and Company, 1981.

[2] E. H. Adelson. Saturation and adaptation in the rod system. *Vision Research*, 22:1299–1312, 1982.

[3] Adobe. Adobe PhothoShop. *http://www.adobe.com/it/products/photoshop/photoshop/*, December 2008.

[4] S. Agarwal, R. Ramamoorthi, S. Belongie, and H. W. Jensen. Structured importance sampling of environment maps. *ACM Transaction on Graphics*, 22(3):605–612, 2003.

[5] A. Agarwala, M. Dontcheva, M. Agrawala, S. Drucker, A. Colburn, B. Curless, D. Salesin, and M. Cohen. Interactive digital photomontage. In *SIGGRAPH '04: ACM SIGGRAPH 2004 Papers*, pages 294–302, New York, NY, USA, 2004. ACM.

[6] T. Akenine-Moller, , and E. Haines. *Real-Time Rendering*. A. K. Peters, Ltd., Natick, MA, USA, 2002.

[7] A. O. Akyüz, R. Fleming, B. E. Riecke, E. Reinhard, and H. H. Bülthoff. Do HDR displays support LDR content?: a psychophysical evaluation. In *SIGGRAPH '07: ACM SIGGRAPH 2007 papers*, page 38, New York, NY, USA, 2007. ACM.

[8] A. O. Akyüz and E. Reinhard. Color appearance in high-dynamic-range imaging. *Journal of Electronic Imaging*, 15(3), July-September 2006.

[9] A. O. Akyüz and E. Reinhard. Perceptual evaluation of tone-reproduction operators using the Cornsweet–Craik–O'brien illusion. *ACM Transaction on Applied Perception*, 4(4):1–29, 2008.

[10] M. Ashikhmin. A tone mapping algorithm for high contrast images. In *EGRW '02: Proceedings of the 13th Eurographics workshop on Rendering*, pages 145–156, Aire-la-Ville, Switzerland, Switzerland, 2002. Eurographics Association.

[11] M. Ashikhmin and J. Goyal. A reality check for tone-mapping operators. *ACM Transaction on Applied Perception*, 3(4):399–411, 2006.

[12] Autodesk. Maya. December 2008.

[13] T. O. Aydin, R. Mantiuk, K. Myszkowski, and H.-P. Seidel. Dynamic range independent image quality assessment. In *SIGGRAPH '08: ACM SIGGRAPH 2008 papers*, pages 1–10, New York, NY, USA, 2008. ACM.

[14] F. Banterle, K. Debattista, P. Ledda, and A. Chalmers. A gpu-friendly method for high dynamic range texture compression using inverse tone mapping. In *GI '08: Proceedings of graphics interface 2008*, pages 41–48, Toronto, Ontario, Canada, 2008. Canadian Information Processing Society.

[15] F. Banterle, P. Ledda, K. Debattista, A. Artusi, M. Bloj, and A. Chalmers. A psychophysical evaluation of inverse tone mapping techniques. *To Appear in Computer Graphics Forum*, 2009.

[16] F. Banterle, P. Ledda, K. Debattista, and A. Chalmers. Inverse tone mapping. In *GRAPHITE '06: Proceedings of the 4th international conference on Computer graphics and interactive techniques in Australasia and Southeast Asia*, pages 349–356, New York, NY, USA, 2006. ACM.

[17] F. Banterle, P. Ledda, K. Debattista, and A. Chalmers. Expanding low dynamic range videos for high dynamic range applications. In *SCCG '08: Proceedings of the 4th Spring Conference on Computer Graphics*, pages 349–356, New York, NY, USA, 2008. ACM.

[18] F. Banterle, P. Ledda, K. Debattista, A. Chalmers, and M. Bloj. A framework for inverse tone mapping. *The Visual Computer*, 23(7):467–478, 2007.

[19] BBC. BBC HD video gallery. In *http://www.bbcmotiongallery.com/*, December 2008.

[20] M. Bertalmio, L. Vese, G. Sapiro, and S. Osher. Simultaneous structure and texture image inpainting. *IEEE Transactions on Image Processing*, 12(8):882–889, August 2003.

[21] J. F. Blinn and M. E. Newell. Texture and reflection in computer generated images. In *SIGGRAPH '76: Proceedings of the 3rd annual conference on Computer graphics and interactive techniques*, pages 266–266, New York, NY, USA, 1976. ACM.

[22] D. Blythe. The direct3d 10 system. In *SIGGRAPH '06: ACM SIGGRAPH 2006 Papers*, pages 724–734, New York, NY, USA, 2006. ACM Press.

[23] G. J. Braun and M. D. Fairchild. Image lightness rescaling using sigmoidal contrast enhancement functions. *Journal of Electronic Imaging*, 8:380–393, 1999.

[24] P. J. Burt and E. H. Adelson. The Laplacian pyramid as a compact image code. *Readings in computer vision: issues, problems, principles, and paradigms*, pages 671–679, 1987.

[25] M. Čadík, M. Wimmer, L. Neumann, and A. Artusi. Image attributes and quality for evaluation of tone mapping operators. In *Proceedings of the 14th Pacific Conference on Computer Graphics and Applications*, pages 35– 44, Taipei, Taiwan, 2006. National Taiwan University Press.

[26] M. Čadík, M. Wimmer, L. Neumann, and A. Artusi. Evaluation of HDR tone mapping methods using essential perceptual attributes. *Computers & Graphics*, 32:330–349, 2008.

[27] J. Carmack. Id tech4: Megatexture. December 2008.

[28] E. E. Catmull. *A subdivision algorithm for computer display of curved surfaces.* PhD thesis, 1974.

[29] S.-F. Chang, T. Sikora, and A. Purl. Overview of the MPEG-7 standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 11(6):688–695, June 2001.

[30] J. Chen, S. Paris, and F. Durand. Real-time edge-aware image processing with the bilateral grid. In *SIGGRAPH '07: ACM SIGGRAPH 2007 papers*, page 103, New York, NY, USA, 2007. ACM.

[31] K. Chiu, M. Herf, P. Shirley, S. Swamy, C. Wang, and K. Zimmerman. Spatially nonuniform scaling functions for high contrast images. In *Proceedings of Graphics Interface 93*, pages 245–244, May 1993.

[32] P. Choudhury and J. Tumblin. The trilateral filter for high contrast images and meshes. In *EGRW '03: Proceedings of the 14th Eurographics workshop on Rendering*, pages 186–196, Aire-la-Ville, Switzerland, Switzerland, 2003. Eurographics Association.

[33] C. Christopoulos, A. Skodras, and T. Ebrahimi. The JPEG2000 still image coding system: an overview. *IEEE Transactions on Consumer Electronics*, 46(4):1103–1127, November 2000.

[34] CIE. Commission Internationale de l'Eclairage. *http://www.cie.co.at*, December 2008.

[35] T. Cornsweet. *Visual perception*. Academic Press, New York, 1970.

[36] M. Corsini, M. Callieri, and P. Cignoni. Stereo light probe. *Computer Graphics Forum*, 27(2):291–300, 2008.

[37] F. C. Crow. Summed-area tables for texture mapping. In *SIGGRAPH '84: Proceedings of the 11th annual conference on Computer graphics and interactive techniques*, pages 207–212, New York, NY, USA, 1984. ACM.

[38] Crytek. Crysis. *http://www.crysis-game.com/*, December 2008.

[39] CypressSemiconductor. LUPA 1300-2. *http://www.cypress.com/*, December 2008.

[40] S. Daly. The visible differences predictor: an algorithm for the assessment of image fidelity. *Digital images and human vision*, pages 179–206, 1993.

[41] S. Daly and X. Feng. Bit-depth extension using spatiotemporal microdither based on models of the equivalent input noise of the visual system. In *Proceedings of Color Imaging VIII: Processing, Hardcopy, and Applications*, volume 5008, 455. SPIE, June 2003.

[42] S. Daly and X. Feng. Decontouring: prevention and removal of false contour artifacts. In *Proceedings of Human Vision and Electronic Imaging IX*, volume 5008, 455. SPIE, June 2004.

[43] H. A. David. *The Method of Paired Comparisons, 2nd ed.* Oxford University Press, 1988.

[44] P. Debevec. Rendering synthetic objects into real scenes: bridging traditional and image-based graphics with global illumination and high dynamic range photography. In *SIGGRAPH '98: Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pages 189–198, New York, NY, USA, 1998. ACM.

[45] P. Debevec. A median cut algorithm for light probe sampling. In *SIGGRAPH '05: ACM SIGGRAPH 2005 Posters*, page 66, New York, NY, USA, 2005. ACM.

[46] P. Debevec and E. Reinhard. High-dynamic-range imaging: theory and applications, 2006.

[47] P. E. Debevec and J. Malik. Recovering high dynamic range radiance maps from photographs. In *SIGGRAPH '97: Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 369–378, New York, NY, USA, 1997. ACM Press/Addison-Wesley Publishing Co.

[48] P. Didyk, R. Mantiuk, M. Hein, and H.-P. Seidel. Enhancement of bright video features for HDR displays. In *Proceeding of Eurographics Symposium on Rendering 2008*, Computer Graphics Forum. Eurographics, Blackwell Ltd, 2008.

[49] Dolby. Dolby-DR37P. *http://www.dolby.com/promo/hdr/technology.html*, December 2008.

[50] F. Drago, W. Martens, K. Myszkowski, and N. Chiba. Design of a tone mapping operator for high dynamic range images based upon psychophysical evaluation and preference mapping. In B. Rogowitz and T. Pappas, editors, *Human Vision and Electronic Imaging VIII (HVEI-03)*, volume 5007 of *SPIE proceedings*, pages 321–331, Santa Clara, USA, April 2003. SPIE.

[51] F. Drago, W. Martens, K. Myszkowski, and H.-P. Seidel. Perceptual evaluation of tone mapping operators with regard to similarity and preference. Research Report MPI-I-2002-4-002, Max-Planck-Institut für Informatik, Stuhlsatzenhausweg 85, 66123 Saarbrücken, Germany, August 2002.

[52] F. Drago, K. Myszkowski, T. Annen, and N. Chiba. Adaptive logarithmic mapping for displaying high contrast scenes. In P. Brunet and D. W. Fellner, editors, *Proc. of EUROGRAPHICS 2003*, volume 22 of *Computer Graphics Forum*, pages 419–426, Granada, Spain, 2003. Blackwell.

[53] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification 2nd Edition*. Wiley Interscience, 2001.

[54] F. Durand and J. Dorsey. Interactive tone mapping. In *Proceedings of the Eurographics Workshop on Rendering Techniques 2000*, pages 219–230, London, UK, 2000. Springer-Verlag.

[55] F. Durand and J. Dorsey. Fast bilateral filtering for the display of high-dynamic-range images. In *SIGGRAPH '02: Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, pages 257–266, New York, NY, USA, 2002. ACM.

[56] D. S. Ebert, F. K. Musgrave, D. Peachey, K. Perlin, and S. Worley. *Texturing and Modeling: A Procedural Approach*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2002.

[57] M. Fairchild. The HDR photographic survey. *http://www.cis.rit.edu/fairchild/HDR.html*, December 2008.

[58] M. D. Fairchild. *Color Appearance Models 2nd Edition*. Wiley-IS&T, 2005.

[59] M. D. Fairchild and G. M. Johnson. Meet icam: A next-generation color appearance model. In *The Tenth Color Imaging Conference*, pages 33–38. IS&T - The Society for Imaging Science and Technology, 2002.

[60] H. Farid. Blind inverse gamma correction. *IEEE Transactions on Image Processing*, 10(10):1428–1433, October 2001.

[61] R. Fattal, D. Lischinski, and M. Werman. Gradient domain high dynamic range compression. In *SIGGRAPH '02: Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, pages 249–256, New York, NY, USA, 2002. ACM.

[62] R. Fedkiw. Ron fedkiw's video gallery. December 2008.

[63] J. A. Ferwerda, S. N. Pattanaik, P. Shirley, and D. P. Greenberg. A model of visual adaptation for realistic image synthesis. In *SIGGRAPH '96: Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 249–258, New York, NY, USA, 1996. ACM.

[64] J. A. Ferwerda, P. Shirley, S. N. Pattanaik, and D. P. Greenberg. A model of visual masking for computer graphics. In *SIGGRAPH '97: Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 143–152, New York, NY, USA, 1997. ACM Press/Addison-Wesley Publishing Co.

[65] R. Feynman, R. Leighton, M. Sands, and E. Hafner. *The Feynman Lectures on Physics; Vol. II*. Addison Wesley, 2005.

[66] J. D. Foley, A. van Dam, S. K. Feiner, and J. F. Hughes. *Computer Graphics: Principles and Practice in C (2nd Edition)*. Addison-Wesley, August 1995.

[67] B. Funt, F. Ciurea, and J. McCann. Retinex in Matlab. In *Proceedings of the IS&T/SID Eighth Color Imaging Conference: Color Science, Systems and Applications*, pages 112–121, 2000.

[68] A. Gilchrist, C. Kossyfidis, F. Bonato, T. Agostini, X. L. Joseph Cataliotti, B. Spehar, V. Annan, and E. Economou. An anchoring theory of lightness perception. *Psychological Review*, 106(4):795–834., October 1999.

[69] R. C. Gonzalez and R. E. Woods. *Digital Image Processing*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2001.

[70] S. J. Gortler, R. Grzeszczuk, R. Szeliski, and M. F. Cohen. The lumigraph. In *SIGGRAPH '96: Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 43–54, New York, NY, USA, 1996. ACM.

[71] N. Greene. Environment mapping and other applications of world projections. *IEEE Computer Graphics and Applications*, 6(11):21–29, November 1986.

[72] M. Grossberg and S. Nayar. Modeling the Space of Camera Response Functions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(10):1272–1282, October 2004.

[73] S. G. Group. The Stanford 3D scanning repository. *http://graphics.stanford.edu/data/3Dscanrep/*, December 2008.

[74] A. Hanjalic. Shot-boundary detection: Unraveled and resolved? *IEEE Transaction Circuits System Video Technology, Vol.12, No.2*, February 2002.

[75] J. H. V. Hateren. Encoding of high dynamic range video with a model of human cones. *ACM Transaction on Graphics*, 25(4):1380–1399, 2006.

[76] J. H. V. Hateren and T. D. Lamb. The photocurrent response of human cones is fast and monophasic. *BMC Neuroscience*, 7(34), April 2006.

[77] T. Hawkins, J. Cohen, and P. Debevec. A photometric approach to digitizing cultural artifacts. In *VAST '01: Proceedings of the 2001 conference on Virtual reality, archeology, and cultural heritage*, pages 333–342, New York, NY, USA, 2001. ACM.

[78] M. M. Hayhoe, N. I. Benimoff, and D. C. Hood. The time course of multiplicative and subtractive adaptation process. *Vision Research*, 27:1981–1996, 1987.

[79] D. Healy and O. Mitchell. Digital video bandwidth compression using block truncation coding. *IEEE Transactions on Communications*, 29(12):1809–1817, December 1981.

[80] P. Heckbert. Color image quantization for frame buffer display. In *SIGGRAPH '82: Proceedings of the 9th annual conference on Computer graphics and interactive techniques*, pages 297–307, New York, NY, USA, 1982. ACM.

[81] B. K. Horn. Determining lightness from an image. *Computer Graphics and Image Processing*, 3(1):277–299, December 1974.

[82] D. Hough. Applications of the proposed IEEE-754 standard for floating point arithmetic. *Computer*, 14(3):70–74, March 1981.

[83] E. Howlett. Wide-angle orthostereo. In J. O. Merritt and S. S. Fisher, editors, *Stereoscopic Displays and Applications*, volume 1256 of *SPIE proceedings*, Santa Clara, USA, February 1990. SPIE.

[84] R. W. G. Hunt. *The Reproduction of Colour*. Fountain Press Ltd, England, 1995.

[85] IDSoftware. Rage. *http://www.idsoftware.com*, to appear in 2009.

[86] Industrial Light & Magic. OpenEXR. *http://www.openexr.org*, December 2008.

[87] K. Iourcha, K. Nayak, and Z. Hong. System and method for fixed-rate block-based image compression with inferred pixel values. *United States Patent 5,956,431*, 1997.

[88] P. Irawan, J. A. Ferwerda, and S. R. Marschner. Perceptually based tone mapping of high dynamic range image streams. In O. Deussen, A. Keller, K. Bala, P. Dutr, D. W. Fellner, and S. N. Spencer, editors, *In Proceedings of the 16th Eurographics Symposium on Rendering*, pages 231–242. Eurographics Association, 2005.

[89] ITU. ITU-R BT.709, basic parameter values for the HDTV standard for the studio and for international programme exchange. In *Standard Recommendation 709, International Telecommunication Union.*, 1990.

[90] H. W. Jensen and P. H. Christensen. Efficient simulation of light transport in scences with participating media using photon maps. In *SIGGRAPH '98: Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pages 311–320, New York, NY, USA, 1998. ACM.

[91] J. T. Kajiya. The rendering equation. *SIGGRAPH Computer Graphics*, 20(4):143–150, 1986.

[92] S. B. Kang, M. Uyttendaele, S. Winder, and R. Szeliski. High dynamic range video. In *SIGGRAPH '03: ACM SIGGRAPH 2003 Papers*, pages 319–325, New York, NY, USA, 2003. ACM.

[93] M. Kendall. *Rank Correlation Methods, 4th ed.* Griffin Ltd., 1975.

[94] J. Kessenich, D. Baldwin, and R. Rost. The OpenGL shading language v.1.20 revision 8. *http://www.opengl.org/documentation/glsl/*, September 2006.

[95] M. Kilgard, P. Brown, and J. Leech. GL_EXT_texture_shared_exponent. In *OpenGL Extension*. http://www.opengl.org/registry/specs/EXT/texture_shared_exponent.txt, 2007.

[96] G. Knittel, A. Schilling, A. Kugler, and W. Straer. Hardware for superior texture performance. *Computers & Graphics*, 20(4):475–481, 1996.

[97] T. Kollig and A. Keller. Efficient illumination by high dynamic range images. In *EGRW '03: Proceedings of the 14th Eurographics workshop on Rendering*, pages 45–50, Aire-la-Ville, Switzerland, Switzerland, 2003. Eurographics Association.

[98] J. Kopf, M. F. Cohen, D. Lischinski, and M. Uyttendaele. Joint bilateral upsampling. In *SIGGRAPH '07: ACM SIGGRAPH 2007 papers*, page 96, New York, NY, USA, 2007. ACM.

[99] G. Krawczyk, K. Myszkowski, and H.-P. Seidel. Lightness perception in tone reproduction for high dynamic range images. In *The European Association for Computer Graphics 26th Annual Conference EUROGRAPHICS 2005*, volume 24 of *Computer Graphics Forum*, pages 635–645, Dublin, Ireland, 2005. Blackwell.

[100] G. Krawczyk, K. Myszkowski, and H.-P. Seidel. Contrast restoration by adaptive countershading. In *The European Association for Computer Graphics Annual Conference EUROGRAPHICS 2007*, volume 26 of *Computer Graphics Forum*. Blackwell, 2007.

[101] J. Kuang, G. M. Johnson, and M. D. Fairchild. icam06: A refined image appearance model for HDR image rendering. *J. Vis. Comun. Image Represent.*, 18(5):406–414, 2007.

[102] J. Kuang, H. Yamaguchi, G. Johnson, and M. Fairchild. Testing HDR image rendering algorithms. In *IS&T/SID 12th Color Imaging Conference*, pages 315–320. SPIE, 2004.

[103] J. Kuang, H. Yamaguchi, C. Liu, G. M. Johnson, and M. D. Fairchild. Evaluating HDR rendering algorithms. *ACM Transaction on Applied Perception*, 4(2):9, 2007.

[104] E. Land. Recent advances in retinex theory. *Vision Research*, 19(1):7–21, 1986.

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

[106] G. W. Larson. Logluv encoding for full-gamut, high-dynamic range images. *Journal of Graphics Tools*, 3(1):15–31, 1998.

[107] 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*, 3(4):291–306, 1997.

[108] P. Ledda, A. Chalmers, T. Troscianko, and H. Seetzen. Evaluation of tone mapping operators using a high dynamic range display. In *SIGGRAPH '05: ACM SIGGRAPH 2005 Papers*, pages 640–648, New York, NY, USA, 2005. ACM.

[109] P. Ledda, L. P. Santos, and A. Chalmers. A local model of eye adaptation for high dynamic range images. In *AFRIGRAPH '04: Proceedings of the 3rd international conference on Computer graphics, virtual reality, visualisation and interaction in Africa*, pages 151–160, New York, NY, USA, 2004. ACM.

[110] P. Ledda, G. Ward, and A. Chalmers. A wide field, high dynamic range, stereographic viewer. In *GRAPHITE '03: Proceedings of the 1st international conference on Computer graphics and interactive techniques in Australasia and South East Asia*, pages 237–244, New York, NY, USA, 2003. ACM.

[111] J.-S. Lee, Y.-Y. Jung, B.-S. Kim, and S.-J. Ko. An advanced video camera system with robust af, ae, and awb control. *IEEE Transactions on Consumer Electronics*, 47(3):694–699, August 2001.

[112] A. Levin, D. Lischinski, and Y. Weiss. Colorization using optimization. In *SIGGRAPH '04: ACM SIGGRAPH 2004 Papers*, pages 689–694, New York, NY, USA, 2004. ACM.

[113] M. Levoy and P. Hanrahan. Light field rendering. In *SIGGRAPH '96: Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 31–42, New York, NY, USA, 1996. ACM.

[114] S. Li. Real-time spherical stereo. In *ICPR '06: Proceedings of the 18th International Conference on Pattern Recognition*, pages 1046–1049, Washington, DC, USA, 2006. IEEE Computer Society.

[115] Y. Li, L. Sharan, and E. H. Adelson. Compressing and companding high dynamic range images with subband architectures. *ACM Transaction on Graphics*, 24(3):836–844, 2005.

[116] S. Lin, J. Gu, S. Yamazaki, and H.-Y. Shum. Radiometric calibration from a single image. In *CVPR 2004: Proceedings of the 2004 IEEE Conference on Computer Vision*

and Pattern Recognition (CVPR2004) - Volume 2, pages 938–945, Washington, DC, USA, 2004. IEEE Computer Society.

[117] S. Lin and L. Zhang. Determining the radiometric response function from a single grayscale image. In *CVPR '05: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 2*, pages 66–73, Washington, DC, USA, 2005. IEEE Computer Society.

[118] D. Lischinski, Z. Farbman, M. Uyttendaele, and R. Szeliski. Interactive local adjustment of tonal values. In *SIGGRAPH '06: ACM SIGGRAPH 2006 Papers*, pages 646–653, New York, NY, USA, 2006. ACM.

[119] S. Lloyd. Least squares quantization in pcm. *IEEE Transactions on Information Theory*, 28(2):129–137, March 1982.

[120] J. Lubin. chapter A Visual Discrimination Model for Imaging System Design and Evaluation, pages 245–283. World Scientific Publishers, 1995.

[121] M. Lyons. Max Lyons's HDR images gallery. *http://www.tawbaware.com/maxlyons/*, December 2008.

[122] T. Malzbender, D. Gelb, and H. Wolters. Polynomial texture maps. In *SIGGRAPH '01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 519–528, New York, NY, USA, 2001. ACM.

[123] S. Mann and R. W. Picard. Being "undigital" with digital cameras: Extending dynamic range by combining differently exposed pictures. In *In Proceedings of IS&T 46th annual conference*, pages 422–428, May 1995.

[124] R. Mantiuk, S. Daly, K. Myszkowski, and H.-P. Seidel. Predicting visible differences in high dynamic range images - model and its calibration. In B. E. Rogowitz, T. N. Pappas, and S. J. Daly, editors, *Human Vision and Electronic Imaging X, IST SPIE's 17th Annual Symposium on Electronic Imaging*, volume 5666, pages 204–214, 2005.

[125] R. Mantiuk, A. Efremov, K. Myszkowski, and H.-P. Seidel. Backward compatible high dynamic range MPEG video compression. In *SIGGRAPH '06: ACM SIGGRAPH 2006 Papers*, pages 713–723, New York, NY, USA, 2006. ACM.

[126] R. Mantiuk, G. Krawczyk, K. Myszkowski, and H.-P. Seidel. Perception-motivated high dynamic range video encoding. In *SIGGRAPH '04: ACM SIGGRAPH 2004 Papers*, pages 733–741, New York, NY, USA, 2004. ACM.

[127] R. Mantiuk, K. Myszkowski, and H.-P. Seidel. Visible difference predicator for high dynamic range images. In *Proceedings of IEEE International Conference on Systems, Man and Cybernetics*, pages 2763–2769, 2004.

[128] W. R. Mark, R. S. Glanville, K. Akeley, and M. J. Kilgard. Cg: a system for programming graphics hardware in a c-like language. In *SIGGRAPH '03: ACM SIGGRAPH 2003 Papers*, pages 896–907, New York, NY, USA, 2003. ACM Press.

[129] G. Mather. *Foundations of Perception 1st edition*. Psychology Press, March 2006.

[130] J. Mendel. Tutorial on higher-order statistics (spectra) in signal processingand system theory: theoretical results and some applications. *Proceedings of the IEEE*, 79(3):278–305, March 1991.

[131] T. Mertens, J. Kautz, and F. V. Reeth. Exposure fusion. In *PG '07: Proceedings of the 15th Pacific Conference on Computer Graphics and Applications*, pages 382–390, Washington, DC, USA, 2007. IEEE Computer Society.

[132] L. Meylan, S. Daly, and S. Ssstrunk. The Reproduction of Specular Highlights on High Dynamic Range Displays. In *IST/SID 14th Color Imaging Conference*, 2006.

[133] L. Meylan, S. Daly, and S. Süsstrunk. Tone Mapping For High Dynamic Range Displays. In *Electronic Imaging*, volume 6492, 2007.

[134] L. Meylan and S. Susstrunk. High dynamic range image rendering with a retinex-based adaptive filter. *IEEE Transactions on Image Processing*, 15(9):2820–2830, 2006.

[135] Microsoft. HLSL: High Level Shading Language. In *Direct3D9c*. Microsoft Corporation, 2003–2008.

[136] . Miller and C. Hoffman. Illumination and reflection maps: Simulated objects in simulated and real environments. In *Siggraph '84 Advanced Computer Graphics Animation seminar note*, New York, NY, USA, July 1984. ACM Press.

[137] T. Mitsunaga and S. Nayar. Radiometric Self Calibration. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 374–380, June 1999.

[138] N. Moroney, M. D. Fairchild, R. W. G. Hunt, C. Li, M. R. Luo, and T. Newman. The ciecam02 color appearance model. In *IS&T/SID 10th Color Imaging Conference*, pages 23–27, 2002.

[139] J. Munkberg, P. Clarberg, J. Hasselgren, and T. Akenine-Möller. High dynamic range texture compression for graphics hardware. *ACM Transaction on Graphics*, 25(3):698–706, 2006.

[140] K. I. Naka and W. A. H. Rushton. S-potentials from luminosity units in the retina of fish (cyprinidae). *Journal of Physiology*, (185):587–599, 1966.

[141] S. Nayar and V. Branzoi. Adaptive Dynamic Range Imaging: Optical Control of Pixel Exposures over Space and Time. In *IEEE International Conference on Computer Vision (ICCV)*, volume 2, pages 1168–1175, October 2003.

[142] L. Neumann, K. Matkovic, and W. Purgathofer. Automatic exposure in computer graphics based on the minimum information loss principle. In *CGI '98: Proceedings of the Computer Graphics International 1998*, page 666, Washington, DC, USA, 1998. IEEE Computer Society.

[143] NeuriCam. NC1805 - Pupilla. *http://www.neuricam.com/*, December 2008.

[144] NextLimits. Maxwell render. *http://www.maxwellrender.com/*, December 2008.

[145] NVIDIA. NVIDIA CUDA compute unified device architecture programming guide. *http://developer.download.nvidia.com/compute/cuda/*, February 2007.

[146] M. Okuda and N. Adami. Two-layer coding algorithm for high dynamic range images based on luminance compensation. *J. Vis. Comun. Image Represent.*, 18(5):377–386, 2007.

[147] Omrom. FZ3 series. *http://www.ia.omron.com/*, December 2008.

[148] V. Ostromoukhov, C. Donohue, and P.-M. Jodoin. Fast hierarchical importance sampling with blue noise properties. In *SIGGRAPH '04: ACM SIGGRAPH 2004 Papers*, pages 488–495, New York, NY, USA, 2004. ACM.

[149] J. D. Owens, D. Luebke, N. Govindaraju, M. Harris, J. Krger, A. E. Lefohn, and T. J. Purcell. A survey of general-purpose computation on graphics hardware. *Computer Graphics Forum*, 26(1):80–113, 2007.

[150] Panoscan. Panoscan MK-3. *http://www.panoscan.com/*, December 2008.

[151] C. M. Papadimitriou. *Computational Complexity*. Addison-Wesley, 1994.

[152] S. Paris and F. Durand. A fast approximation of the bilateral filter using a signal processing approach. In *ECCV (4)*, pages 568–580, 2006.

[153] S. Paris, P. Kornprobst, J. Tumblin, and F. Durand. A gentle introduction to bilateral filtering and its applications. In *SIGGRAPH '07: ACM SIGGRAPH 2007 courses*, page 1, New York, NY, USA, 2007. ACM.

[154] S. Pattanaik and H. Yee. Adaptive gain control for high dynamic range image display. In *SCCG '02: Proceedings of the 18th spring conference on Computer graphics*, pages 83–87, New York, NY, USA, 2002. ACM.

[155] S. N. Pattanaik, J. A. Ferwerda, M. D. Fairchild, and D. P. Greenberg. A multiscale model of adaptation and spatial vision for realistic image display. In *SIGGRAPH '98: Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pages 287–298, New York, NY, USA, 1998. ACM.

[156] S. N. Pattanaik, J. Tumblin, H. Yee, and D. P. Greenberg. Time-dependent visual adaptation for fast realistic image display. In *SIGGRAPH '00: Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 47–54, New York, NY, USA, 2000. ACM Press/Addison-Wesley Publishing Co.

[157] E. S. Pearson and H. O. Hartley. *Biometrika tables for statisticians 3rd ed., vol. 1*. Cambridge University Press, 1988.

[158] F. Pellacini, J. A. Ferwerda, and D. P. Greenberg. Toward a psychophysically-based light reflection model for image synthesis. In *SIGGRAPH '00: Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 55–64, New York, NY, USA, 2000. ACM Press/Addison-Wesley Publishing Co.

[159] P. Pérez, M. Gangnet, and A. Blake. Poisson image editing. *ACM Transaction on Graphics*, 22(3):313–318, 2003.

[160] G. Petschnigg, R. Szeliski, M. Agrawala, M. Cohen, H. Hoppe, and K. Toyama. Digital photography with flash and no-flash image pairs. In *SIGGRAPH '04: ACM SIGGRAPH 2004 Papers*, pages 664–672, New York, NY, USA, 2004. ACM.

[161] T. Pham and L. van Vliet. Separable bilateral filtering for fast video preprocessing. In *IEEE International Conference on Multimedia and Expo, 2005. ICME 2005*, October 2005.

[162] M. Pharr and G. Humphreys. *Physically Based Rendering: From Theory to Implementation*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2004.

[163] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes 3rd Edition: The Art of Scientific Computing*. Cambridge University Press, September 2007.

[164] PtGreyResearch. Firefly MV. *http://www.ptgrey.com/*, December 2008.

[165] Z. Rahman, D. Jobson, and G. Woodell. Multi-scale retinex for color image enhancement. In *Proceedings of the International Conference on Image Processing*, volume 3, pages 1003–1006, 1996.

[166] G. Ramanarayanan, J. Ferwerda, B. Walter, and K. Bala. Visual equivalence: towards a new standard for image fidelity. In *SIGGRAPH '07: ACM SIGGRAPH 2007 papers*, page 76, New York, NY, USA, 2007. ACM.

[167] RedCompany. Read One. *http://www.red.com/*, December 2008.

[168] E. Reinhard. Parameter estimation for photographic tone reproduction. *Journal Graphics Tools*, 7(1):45–52, 2002.

[169] E. Reinhard, M. Stark, P. Shirley, and J. Ferwerda. Photographic tone reproduction for digital images. In *SIGGRAPH '02: Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, pages 267–276, New York, NY, USA, 2002. ACM.

[170] E. Reinhard, G. Ward, S. Pattanaik, and P. Debevec. *High Dynamic Range Imaging: Acquisition, Display and Image-Based Lighting*. Morgan Kaufmann Publishers, December 2005.

[171] A. G. Rempel, M. Trentacoste, H. Seetzen, H. D. Young, W. Heidrich, L. Whitehead, and G. Ward. Ldr2Hdr: on-the-fly reverse tone mapping of legacy video and photographs. In *SIGGRAPH '07: ACM SIGGRAPH 2007 papers*, New York, NY, USA, 2007. ACM Press.

[172] R. L. Rivest and C. E. Leiserson. *Introduction to Algorithms*. McGraw-Hill, Inc., New York, NY, USA, 1990.

[173] L. Roberts. Picture coding using pseudo-random noise. *IEEE Transactions on Information Theory*, 8(2):145–154, February 1962.

[174] K. Roimela, T. Aarnio, and J. Itäranta. High dynamic range texture compression. In *SIGGRAPH '06: ACM SIGGRAPH 2006 Papers*, pages 707–712, New York, NY, USA, 2006. ACM.

[175] K. Roimela, T. Aarnio, and J. Itäranta. Efficient high dynamic range texture compression. In *SI3D '08: Proceedings of the 2008 symposium on Interactive 3D graphics and games*, pages 207–214, New York, NY, USA, 2008. ACM.

[176] A. I. Ruppertsberg, M. Bloj, F. Banterle, and A. Chalmers. Displaying colourimetrically calibrated images on a high dynamic range display. *Journal Visual Communication Image Represent.*, 18(5):429–438, 2007.

[177] H. Samet. *Foundations of Multidimensional and Metric Data Structures (The Morgan Kaufmann Series in Computer Graphics and Geometric Modeling)*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2005.

[178] C. Schlick. Quantization techniques for visualization of high dynamic range pictures. In *Proceeding of the Fifth Eurographics Workshop on Rendering*, pages 7–18, June 1994.

[179] H. Seetzen, W. Heidrich, W. Stuerzlinger, G. Ward, L. Whitehead, M. Trentacoste, A. Ghosh, and A. Vorozcovs. High dynamic range display systems. *ACM Transaction on Graphics*, 23(3):760–768, 2004.

[180] S. Siegel and N. J. Castellan. *Nonparametric Statistics for the Behavioral Sciences*. McGrall-Hill International, 1988.

[181] K. Smith, G. Krawczyk, K. Myszkowski, and H.-P. Seidel. Beyond tone mapping: Enhanced depiction of tone mapped HDR images. In E. Gröller and L. Szirmay-Kalos, editors, *EUROGRAPHICS 2006 (EG'06)*, volume 25 of *Computer Graphics Forum*, pages 427–438, Vienna, Austria, September 2006. Eurographics, Blackwell.

[182] Spheron. Spheron HDR VR. *http://www.spheron.com/*, December 2008.

[183] J. Stevens and S. Stevens. Brightness function: Effects of adaptation. *Journal Optical Society of America*, 53(3):375–385, March 1963.

[184] S. Stevens and J. Stevens. Brightness function: Parametric effects of adaptation and contrast. *Journal Optical Society of America*, 50(11), November 1960.

[185] M. Stokes, M. Anderson, S. Chandrasekar, and R. Motta. A Standard Default Color Space for the Internet - sRGB. *World Wide Web Consortium*, November 1996.

[186] E. J. Stollnitz, T. D. DeRose, and D. H. Salesin. Wavelets for computer graphics: A primer. *IEEE Comput. Graph. Appl.*, 15(3):76–84, 1995.

[187] J. Stumpfel, C. Tchou, A. Jones, T. Hawkins, A. Wenger, and P. Debevec. Direct HDR capture of the sun and sky. In *AFRIGRAPH '04: Proceedings of the 3rd international conference on Computer graphics, virtual reality, visualisation and interaction in Africa*, pages 145–149, New York, NY, USA, 2004. ACM.

[188] C. Tchou, J. Stumpfel, P. Einarsson, M. Fajardo, and P. Debevec. Unlighting the parthenon. In *SIGGRAPH '04: ACM SIGGRAPH 2004 Sketches*, page 80, New York, NY, USA, 2004. ACM.

[189] ThomsonGrassValley. Viper FilmStream. *http://www.thomsongrassvalley.com/*, December 2008.

[190] L. L. Thurstone. A law of comparative judgment. In *Psychological Review 34*, pages 273–286, 1927.

[191] C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. In *ICCV '98: Proceedings of the Sixth International Conference on Computer Vision*, page 839, Washington, DC, USA, 1998. IEEE Computer Society.

[192] M. Trentacoste, W. Heidrich, L. Whitehead, H. Seetzen, and G. Ward. Photometric image processing for high dynamic range displays. *Journal Visual Communication Image Represent.*, 18(5):439–451, 2007.

[193] J. Tumblin, J. K. Hodgins, and B. K. Guenter. Two methods for display of high contrast images. *ACM Transaction on Graphics*, 18(1):56–94, 1999.

[194] J. Tumblin and H. Rushmeier. Tone reproduction for realistic images. *IEEE Comput. Graph. Appl.*, 13(6):42–48, 1993.

[195] J. Tumblin and G. Turk. Lcis: a boundary hierarchy for detail-preserving contrast reduction. In *SIGGRAPH '99: Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 83–90, New York, NY, USA, 1999. ACM Press/Addison-Wesley Publishing Co.

[196] J. Unger, S. Gustavson, P. Larsson, and A. Ynnerman. Free form incident light fields. *Computer Graphics Forum*, 27(4), June 2008.

[197] J. Unger, A. Wenger, T. Hawkins, A. Gardner, and P. Debevec. Capturing and rendering with incident light fields. In *EGRW '03: Proceedings of the 14th Eurographics workshop on Rendering*, pages 141–149, Aire-la-Ville, Switzerland, Switzerland, 2003. Eurographics Association.

[198] J. Valeton and D. van Norren. Light adaptation of primate cones: an analysis based on extracellular data. *Vision Research*, 23(12):1539–1547, 1983.

[199] J. van Waveren. Real-Time DXT Compression. *Intel Software Network*, October 2006.

[200] V. N. Vapnik. *The nature of statistical learning theory*. Springer-Verlag New York, Inc., New York, NY, USA, 1995.

[201] D. Vatolin and A. Noskov. Brightness flicking metric. *http://compression.ru/video/quality_measure/metric_plugins/bfm_en.htm*, 2002.

[202] VisionResearch. Phantom HD. *http://www.visionresearch.com/*, December 2008.

[203] I. Wald, W. R. Mark, J. Günther, S. Boulos, T. Ize, W. Hunt, S. G. Parker, and P. Shirley. State of the art in ray tracing animated scenes. In D. Schmalstieg and J. Bittner, editors, *STAR Proceedings of Eurographics 2007*, pages 89–116. The Eurographics Association, Sept. 2007.

[204] Walraven and Valeton. *Visual adaptation and response saturation*. VNU Science Press, 1984.

[205] S. Wan, M. Mrak, and E. Izquierdo. Perceptually adaptive joint deringing - deblocking filtering for scalable video coding. In *MobiMedia '06: Proceedings of the 2nd international conference on Mobile multimedia communications*, pages 1–5, New York, NY, USA, 2006. ACM.

[206] L. Wang, X. Wang, P.-P. Sloan, L.-Y. Wei, X. Tong, and B. Guo. Rendering from compressed high dynamic range textures on programmable graphics hardware. In *I3D '07: Proceedings of the 2007 symposium on Interactive 3D graphics and games*, pages 17–24, New York, NY, USA, 2007. ACM.

[207] L. Wang, L.-Y. Wei, K. Zhou, B. Guo, and H.-Y. Shum. High dynamic range image hallucination. In *Proceedings of Eurographics Symposium on Rendering*, Jun 2007.

[208] Z. Wang and A. Bovik. A universal image quality index. *IEEE Signal Processing Letters*, 9(3):81–84, March 2002.

[209] G. Ward. Real pixels. *Graphics Gems*, 2:15–31, 1991.

[210] G. Ward. *A Contrast-Based Scalefactor for Luminance Display*. Academic Press, 1994.

[211] G. Ward. A wide field, high dynamic range, stereographic viewer. In *In Proceeding of PICS 2002*, 2002.

[212] G. Ward. Greg Ward's HDR images gallery. *http://www.anyhere.com/gward/*, December 2008.

[213] G. Ward and M. Simmons. Subband encoding of high dynamic range imagery. In *APGV '04: Proceedings of the 1st Symposium on Applied perception in graphics and visualization*, pages 83–90, New York, NY, USA, 2004. ACM Press.

[214] G. Ward and M. Simmons. JPEG-HDR: A backwards-compatible, high dynamic range extension to JPEG,. In *CIC 13th: Proceedings of the Thirteenth Color Imaging Conference*. The Society for Imaging Science and Technology, 2005.

[215] G. J. Ward. Measuring and modeling anisotropic reflection. In *SIGGRAPH '92: Proceedings of the 19th annual conference on Computer graphics and interactive techniques*, pages 265–272, New York, NY, USA, 1992. ACM.

[216] G. J. Ward. The radiance lighting simulation and rendering system. In *SIGGRAPH '94: Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, pages 459–472, New York, NY, USA, 1994. ACM.

[217] A. B. Watson. *Temporal Sensitivity Chapter 6 in Handbook of Perception, Vol. I*. Springer Verlag, 1986.

[218] A. B. Watson and J. Solomon. Model of visual contrast gain control and pattern masking. *Journal of the Optical Society of America*, 14(9):2379–2391, 1997.

[219] T. Whitted. An improved illumination model for shaded display. In *SIGGRAPH '79: Proceedings of the 6th annual conference on Computer graphics and interactive techniques*, page 14, New York, NY, USA, 1979. ACM.

[220] T. Wiegand, G. Sullivan, G. Bjontegaard, and A. Luthra. Overview of the H.264/AVC video coding standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 13(7):560–576, July 2003.

[221] L. Williams. Casting curved shadows on curved surfaces. In *SIGGRAPH '78: Proceedings of the 5th annual conference on Computer graphics and interactive techniques*, pages 270–274, New York, NY, USA, 1978. ACM.

[222] H. R. Wilson. Psychophysical models of spatial vision and hyperacuity. In *Spatial Vision. Vision and Visual Dysfunction*, volume Vol. 10, pages 64–81. Boca Raton, FL, CRC Press, 1991.

[223] R. Xu, S. N. Pattanaik, and C. E. Hughes. High-dynamic-range still-image encoding in jpeg 2000. *IEEE Computer Graphics and Applications*, 25(6):57–64, 2005.

[224] H. Yee and S. Pattanaik. Segmentation and adaptive assimilation for detail-preserving display of high-dynamic range images. *The Visual Computer*, 19(7/8):457–466, December 2003.

[225] A. Yoshida, V. Blanz, K. Myszkowski, and H.-P. Seidel. Perceptual evaluation of tone mapping operators with real-world sceness. In B. E. Rogowitz, T. N. Pappas, and S. J. Daly, editors, *Human Vision and Electronic Imaging X, IS&T/SPIE's 17th Annual Symposium on Electronic Imaging (2005)*, volume 5666 of *SPIE Proceedings Series*, pages 192–203, San Jose, USA, January 2005. SPIE.

[226] A. Yoshida, V. Blanz, K. Myszkowski, and H.-P. Seidel. Testing tone mapping operators with human-perceived reality. *Journal of Electronic Imaging*, 16(1), Jannuary-March 2007.

[227] A. Yoshida, R. Mantiuk, K. Myszkowski, and H.-P. Seidel. Analysis of reproducing real-world appearance on displays of varying dynamic range. *Computer Graphics Forum*, 25(3):415–426, Sept. 2006.