

# TONE MAPPING BY INTERACTIVE EVOLUTION

by

Stephen B. Chisholm

Submitted in partial fulfillment of the requirements  
for the degree of Master of Computer Science

at

Dalhousie University  
Halifax, Nova Scotia  
October 2009

© Copyright by Stephen B. Chisholm, 2009

DALHOUSIE UNIVERSITY

FACULTY OF COMPUTER SCIENCE

The undersigned hereby certify that they have read and recommend to the Faculty of Graduate Studies for acceptance a thesis entitled “TONE MAPPING BY INTERACTIVE EVOLUTION” by Stephen B. Chisholm in partial fulfillment of the requirements for the degree of Master of Computer Science.

Dated: October 8, 2009

Supervisors:

---

---

Reader:

---

DALHOUSIE UNIVERSITY

DATE: October 8, 2009

AUTHOR: Stephen B. Chisholm

TITLE: TONE MAPPING BY INTERACTIVE EVOLUTION

DEPARTMENT OR SCHOOL: Faculty of Computer Science

DEGREE: MSc

CONVOCATION: May

YEAR: 2010

Permission is herewith granted to Dalhousie University to circulate and to have copied for non-commercial purposes, at its discretion, the above title upon the request of individuals or institutions.

---

Signature of Author

The author reserves other publication rights, and neither the thesis nor extensive extracts from it may be printed or otherwise reproduced without the author's written permission.

The author attests that permission has been obtained for the use of any copyrighted material appearing in the thesis (other than brief excerpts requiring only proper acknowledgement in scholarly writing) and that all such use is clearly acknowledged.

# Table of Contents

<b>List of Tables</b> . . . . .	<b>vi</b>
<b>List of Figures</b> . . . . .	<b>vii</b>
<b>Abstract</b> . . . . .	<b>xii</b>
<b>Acknowledgements</b> . . . . .	<b>xiii</b>
<b>Chapter 1 Introduction</b> . . . . .	<b>1</b>
1.1 Tone Mapping . . . . .	1
1.2 High Dynamic Range Images . . . . .	1
1.3 The Problem . . . . .	3
1.4 The Proposed Solution . . . . .	5
<b>Chapter 2 Background</b> . . . . .	<b>8</b>
2.1 Overview . . . . .	8
2.2 High Dynamic Range Image Formats . . . . .	8
2.3 Colour Spaces . . . . .	9
2.3.1 RGB and XYZ . . . . .	11
2.3.2 Decorrelated Colour Space $L\alpha\beta$ . . . . .	13
2.3.3 Perceptually Uniform Colour Space $L^*a^*b^*$ . . . . .	14
2.4 Image Metrics . . . . .	17
2.4.1 Mean Squared Error and Peak Signal to Noise . . . . .	18
2.4.2 Universal Quality Index . . . . .	18
2.4.3 Universal Colour Image Fidelity Metric . . . . .	21
2.5 Tone Mapping Operators . . . . .	22
2.5.1 Schlick's Operator . . . . .	24
2.5.2 Ashikhmin's Operator . . . . .	26
2.5.3 Reinhard and Devlin . . . . .	29

2.6	Interactive Evolution . . . . .	31
2.7	Constraint Handling . . . . .	32
<b>Chapter 3</b>	<b>Interactive Evolutionary Tone Mapping . . . . .</b>	<b>34</b>
3.1	Basic Algorithm . . . . .	36
3.1.1	Search Space . . . . .	36
3.1.2	Generating Offspring . . . . .	37
3.1.3	Image Blending . . . . .	38
3.2	Perceptually Based Step Length Adaptation . . . . .	39
3.2.1	Perceptual Ranges . . . . .	40
3.2.2	Perceptual Similarity Ranges . . . . .	43
3.2.3	Speed Enhancements . . . . .	45
3.3	Results . . . . .	47
3.3.1	Offspring Step Size Control . . . . .	47
3.3.2	Parameter Space Step Size Adaptation . . . . .	53
3.3.3	Perceptual Space Step Size Adaptation . . . . .	55
<b>Chapter 4</b>	<b>Conclusion . . . . .</b>	<b>61</b>
	<b>Bibliography . . . . .</b>	<b>62</b>

## List of Tables

3.1	The default, minimum, and maximum values which are used for the tone mapping operators' input parameters. Default values are used for the parent of the first generation and the minimum and maximum values are used as constraints. . . . .	37
-----	--	----

## List of Figures

1.1	A series of images taken of the same scene at different exposures, as well as the result of combining the images to a single high dynamic range image which is then tone mapped with the iCAM operator. This series of images is taken from [11]. . . . .	2
1.2	This series of images shows the diversity of the different tone mapping operators. These images are mapped very differently by each of the tone mapping operators using their default parameter settings. By tuning the parameters each operator should be able to give a much improved result. . . . .	3
1.3	The mapping on the left was done with the gradient domain operator of Fattal et al. [14], using the parameters to boost contrast and saturation which leads to more of a surreal unnatural appearance. The mapping on the right on the other hand, published in [14], mapped with the same operator has a much more natural appearance giving the viewer more of an impression of the original scene. . . . .	4
2.1	This plot from [39] shows the colour matching functions for 2° CIEXYZ. . . . .	10
2.2	This example shows how the RGB colour channels are correlated and how $L\alpha\beta$ is able to remove this correlation. In the first row of plots we can see that for the most part for a given contribution of one colour channel a very similar contribution of the other channels can be predicted. Looking at the second row of plots, we see that $L\alpha\beta$ has transformed the data such that it is aligned with and roughly symmetric about the axes. This non-correlated colour space is important in that three dimensional image processing problems can be looked at as three separate one dimensional problems. . . . .	12
2.3	A plot of MacAdam ellipses used to show the non uniformity of XYZ taken from [16]. Each ellipse shows a region of colours which are indistinguishable from one another. These ellipses are enlarged so that they are easier to see. Their overall size is not important here, it is that their size and orientation relative to one another vary over the plot. . . . .	14

2.4	MacAdam ellipses for the CIELUV colour space taken from [16]. This colour space is more perceptually uniform. It is very similar to the CIELAB colour space and the reason that they both exist according to [34] is largely historic. CIELAB is commonly used in print material where CIELUV is more common with television. In this plot we can see that the ellipses are much closer in size and orientation when compared with those in Figure 2.3. . . . . .	16
2.5	This figure compares Mean Squared Error (MSE) with the Structural Similarity Image Metric (SSIM) for different types of distortions. These images come from a similar figure in [46], these images have been distorted such that each have a similar MSE score and this leaves open for interpretation the scores of the SSIM. The distortions are: (a) Original, (b) Contrast Stretch, (c) Luminance Shift, (d) Blurring, (e) Gaussian Noise, (f) JPEG Compression. Here it should also be noted that SSIM has a maximum of 1.0 obtained when the two images being compared are identical. MSE on the other hand has an optimal value of 0.0 as it is a measure of error. . . . . .	19
2.6	This image is provided by Schlick in [36] for calculating the darkest grey visible on a display device. By viewing this image on the target display device and selecting the first square in the image which is distinguishable from the black background, this gives the intensity which can be use as a value for the parameter $M$ . . . . . .	26
2.7	These plots show some curves representing gamma corrected linear mappings (or exponential), logarithmic mappings and rational mappings. As we can see the rational mappings are able to adopt shapes similar to those of the exponential and logarithmic mappings as well as form further shapes. When viewed over a logarithmic $x$ -axis the rational mappings exhibit a sigmoidal shape. This allows for the extreme ends of the compression range to be treated differently than the rest. . . . . .	27
3.1	A screen shot of the interface. The user is presented with nine images with a diverse selection. There are a few which are beyond anything that might be considered desirable. However, there are several very eligible candidates each with unique characteristics. . . . . .	35



3.2	These plots show the average perceptual similarity of offspring from four different parents at different step sizes. The perceptual similarity has a range of $[0.0, 2.24722]$ and the step size is varied from 0.0 to 1.4 with discrete steps of 0.2. The Paul Bunyan image can be seen in Figure 3.3 and the Waffle House image in Figure 3.11. . . . .	40
3.3	This series of images shows how the similarity metric $Q_{colour}$ behaves with respect to step size. The images here are the result of fixing a single random direction in parameter space and then generating an image at increasing step sizes. In this case we see that the further along the direction we travel away from the image the metric decreases. . . . .	41
3.4	This figure shows the average number of images generated for different perceptual similarity targets and margins of error in order to generate eight child images within the range defined by the target and margin. The offspring for this plot were all generated using an initial step size of 0.2. This was done for four different images from [11]. These results were created to support the decisions regarding the adaptation of the perceptual ranges. . . . .	44
3.5	This shows how the range of perceptual similarities will adapt with each update which occurs. The maximum line shows the maximum value which the similarity metric can obtain. The margin of error and target lines show how each of these are adapted to define the range of similarities. . . . .	45
3.6	This figure shows the relationship between the size of the image and perceptual similarity measured with the universal colour image fidelity metric. Four pairs of images with an initial size of $1024 \times 683$ are compared after being resized from 99% to 1%. The resizing was done using the box filter as it left the similarity metric the least affected. From the plot we can see that until the 5% line all of the curves increase gradually. At this point the images have a resolution of $50 \times 34$ . . . . .	46
3.7	This is the initial generation of images for the sample session using parameter step size adaptation. The image seen here and in the follow three figures is an image of the Golden Gate Bridge found in [11]. . . . .	49

3.8	This is the initial generation of images for the sample session using perceptual space step size adaptation. We can see that there are no images similar to the one found in the centre of the bottom row of Figure 3.7. . . . . .	50
3.9	This is the final generation of images for the sample session using parameter space step size adaptation. We can see that even though the step size has been reduced there is still at least one image which is still fairly different from the parent image in the centre. . . . . .	51
3.10	This is the final generation of image for the sample session using perceptual space adaptation. We can see here that by restricting the images in perceptual space we are able to produce a generation of very similar offspring. . . . . .	52
3.11	A sample session with parameter space step size adaptation using the Waffle House image. The final blended image does a good job at eliminating the bright spots around the sign while giving a good sense of the brightness of the sign. . . . . .	54
3.12	The plot shows how the weights adapt over the generations with parameter space step size adaptation. The places in the plot where all three of the weights do not change between generations indicate areas where the parent was selected. . . . . .	55
3.13	A sample session with perceptual space step size using the Waffle House image. The final blended image is quite similar to the previous sample session however the final blended image is somewhat brighter overall, this is likely due to the larger weighting given to Schlick's operator. . . . . .	56
3.14	This plot shows the mutations of the weights for a sample session similar to the previous except this time with perceptual space step size adaptation. The weights here are very much the same as the previous session except for the weight associated with Schlick's operator which is larger in this session. This could be what attributed to the brighter final blended image. . . . . .	57
3.15	Comparison with other published results. The mapping on the left is a result from Fattal et al. [14] and the mapping on the right is from Ward Larson et al. [22]. The centre mapping was generated with the interactive evolutionary tone mapping tool. The high dynamic image is due to Tumblin and Turk [44]. . . . . .	58

3.16	A second comparison with other published results. The mapping on the left is a result from Durand and Dorsey’s bilateral filtering algorithm [10] and the mapping on the right is from Ward Larson et al. [22]. The centre mapping was generated with the interactive evolutionary tone mapping tool. The high dynamic range image is included with [34]. . . . .	59
3.17	Third and final comparison with other published results. The mappings on the top are two result from Choudhury and Tumblin’s trilateral filtering algorithm [6]. On the bottom are results from Fattal et al. [14] and the mappings in the centre are results from the interactive tone mapping tool. Both of these images are due to Fattal et al. [14]. . . . .	60

## **Abstract**

Tone mapping is a computational task of significance in the context of displaying high dynamic range images on low dynamic range devices. While a number of tone mapping algorithms have been proposed and are in common use, there is no single operator that yields optimal results under all conditions. Moreover, obtaining satisfactory mappings often requires the manual tweaking of parameters. This thesis proposes interactive evolution as a computational tool for tone mapping. An evolution strategy that blends the results from several tone mapping operators while at the same time adapting their parameters is proposed. As well, the results are adapted such that such that approximately uniform perceptual distances between offspring candidate solutions and the parent are ensured. The introduction of a perceptually based step size adaptation technique enhances the control of the variability between newly generated offspring, when compared to parameter space step size adaptation.

## Acknowledgements

Thanks to my family and friends for all of their support and encouragement. As well, thanks to my supervisors for their help and guidance.

# Chapter 1

## Introduction

### 1.1 Tone Mapping

Tone mapping is an important task in image processing and computer graphics. While the luminance present in real world scenes often spans many orders of magnitude, today's display devices, including CRTs, LCDs, and printers, are capable of displaying only a narrow subrange thereof. For example, the contrast ratio, i.e. the ratio of luminance values between the brightest and darkest regions in an image depicting an indoor scene in which a window looking into the sunlit outdoors is visible, can span up to nine orders of magnitude. The contrast ratio that contemporary display devices are capable of generating spans a mere two or three orders of magnitude. The task of mapping the colours of a high dynamic range image to the low dynamic range of a display device such that the visual appearance of the image is preserved as much as possible is referred to as tone mapping. Even as display devices capable of displaying higher dynamic ranges become available, tone mapping will remain an important task as lower cost devices including printers are not likely to disappear. The increasing importance of tone mapping is also witnessed by its growing support in commonly used image processing software. A good introduction to the area of high dynamic range imaging can be found in Reinhard et al. [34].

### 1.2 High Dynamic Range Images

High dynamic range images can stem from several sources. One possibility is to capture several images of the same scene with different exposure settings. This allows various amounts of light to enter the lens. Debevec and Malik [8] and others have developed algorithms for stitching those shots together to form a single high dynamic range image. Clearly, this approach is best suited for still scenes. Figure 1.1 shows

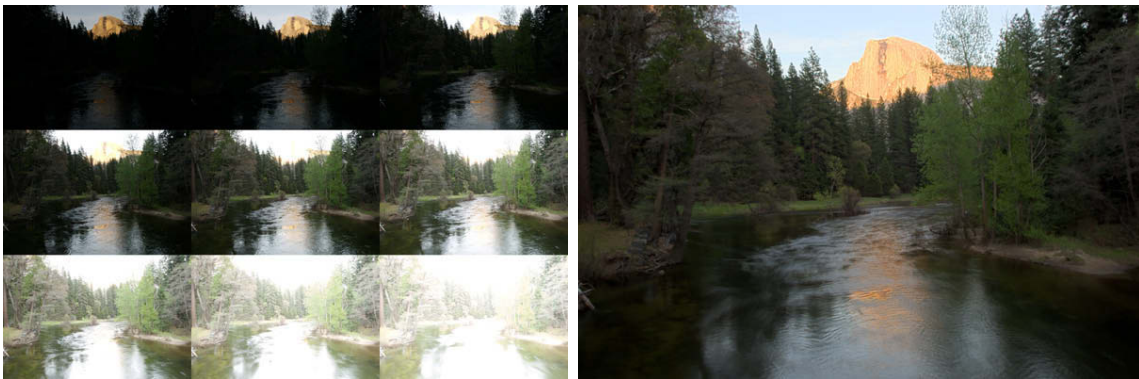


Figure 1.1: A series of images taken of the same scene at different exposures, as well as the result of combining the images to a single high dynamic range image which is then tone mapped with the iCAM operator. This series of images is taken from [11].

an example of this, where on the left we see the series of images taken at different exposures allowing increasingly more light into the lens for each image. This allows for each portion of the scene to be better exposed. On the right we see a tone mapped version of the high dynamic range image which was created as the combination of the series. Notice that the details in the dark areas of the scene, such as the forest around the river, as well as the bright areas around mountain peak are both visible in the tone mapped image. However, looking at the series of images, the details of a particular area are only visible in the single exposure where the area was correctly exposed. A second possibility for acquiring high dynamic range images is through advances in camera technology. Many modern digital cameras that are on the market today already use higher dynamic range representations internally than were available in the past. And finally, photo-realistic rendering techniques that use physically based models of light propagation, such as ray tracing, artificially generate high dynamic range images [29]. While most image formats in common use today represent colours using eight bits in each colour channel (thus being able to represent only 256 distinct values per channel), the use of several high dynamic range image formats that use more bits per pixel, such as OpenEXR, is becoming more widespread.

A multitude of tone mapping operators have been proposed, ranging from the use of simple sigmoidal functions to the computationally and conceptually more sophisticated gradient domain tone mapping operator by Fattal et al. [14]. Some operators



Figure 1.2: This series of images shows the diversity of the different tone mapping operators. These images are mapped very differently by each of the tone mapping operators using their default parameter settings. By tuning the parameters each operator should be able to give a much improved result.

are based on psycho-physical models of human perception; others strive to either locally or globally maximise contrast. Figure 1.2 shows the results of applying three different tone mapping operators to two high dynamic range images. In this figure the results from Schlick's [36] operator produces the darkest image for the first scene, yet for the second image the operator generates an unnaturally bright image. In the former case, Schlick's operator is also the one which preserves the greatest amount of local detail. Ashikhmin's operator as well as Reinhard and Devlin's operator generate quite bright results for the first scene while leaving out most of the details. However, in the second image however both operators generate a much better mapping with many details present. In all cases, better images can be generated by tuning the algorithms' parameters.

### 1.3 The Problem

High dynamic range images and tone mapping have engendered many debates in photography communities on the internet. One interesting side effect of high dynamic range images is that the extra information stored in the image can produce some unique visual effects when tone mapping the image. These give the images a somewhat





Figure 1.3: The mapping on the left was done with the gradient domain operator of Fattal et al. [14], using the parameters to boost contrast and saturation which leads to more of a surreal unnatural appearance. The mapping on the right on the other hand, published in [14], mapped with the same operator has a much more natural appearance giving the viewer more of an impression of the original scene.

surreal appearance. Some operators allow, with the tweaking of parameters, for the creation of mappings which have perhaps an unnatural amount of contrast. This has been exploited in some cases by groups of individuals using HDR to create images with very different effects. A guide to HDR photography [25] demonstrates how to achieve these effects to match these types of images. Figure 1.3 shows an example of this. By boosting contrast and saturation the mapping on the left is quite unnatural compared to the mapping on the right which is mapped to give the image the same impression as the original scene. This is most often the intent of tone mapping operators: striving to produce mappings which if compared to the original scene come as close as possible to reality.

As of today, there is no single “best” tone mapping operator that generates satisfactory results under all conditions. Several recent studies that attempt to compare

different operators arrive at different conclusions. Drago et al. [9] conducted a study in which human subjects assessed their perceptions when comparing the results of six tone mapping operators applied to high dynamic range images of four scenes, including both synthetic and photographic ones. Subjects were asked to judge how perceptually similar or dissimilar the images were with respect to apparent image contrast, apparent level of detail, and apparent naturalness. As a result, they were able to group operators into categories that best preserve each one of the three attributes, albeit usually at the cost of the other two. Ledda et al. [23] had human subjects compare images mapped using six common tone mapping operators with images of the same scenes displayed on a high dynamic range device. They found that subjects tended to agree on which tone mapped images appear closest to the reference image, and that the photographic tone reproduction algorithm of Reinhard et al. [33] and the iCAM operator of Johnson and Fairchild [20] consistently performed well. When asked to rate the reproduction of feature and detail, agreement between subjects was even more pronounced and again seemed to favour the iCAM operator. Kuang et al. [21] conducted further psycho-physical experiments in which they had human subjects compare images tone mapped by different operators with the real world scenes they depict. In their study, the iCAM operator’s performance was rather mixed, and their results seemed to favour the bilateral filtering algorithm of Durand and Dorsey [10] instead. However, they also state that “no single algorithm consistently performs well for all images.” It is unclear whether parameter tuning was done for the operators included in their study.

#### 1.4 The Proposed Solution

Algorithmically, there currently is no conclusive answer to the question of which operator is best suited to tone mapping a given high dynamic range image, and how its parameters should be set. However, a human subject is typically able to effortlessly evaluate the quality of a tone mapped image. We therefore contend that interactive evolution is a computational tool ideally suited to the task of tone mapping. Interactive evolutionary algorithms have previously been used for tasks ranging from the

computer assisted evolution of coffee blends to the fitting of hearing aids. Applications of interactive evolution in image processing and computer graphics include algorithms for interactive lighting design [24], the computer assisted creation of art [40], and the interactive evolution of images [17, 49, 28, 37, 38]. Despite the significance of the task, no evolutionary approaches have been proposed for the problem of tone mapping high dynamic range images.

In this thesis we propose an approach to tone mapping that uses interactive evolutionary techniques. Tone mapped images are generated by blending the results obtained from a number of commonly used tone mapping operators. The weights that determine the relative influence of the individual operators together with the operators' parameters form a vector of real-valued variables. An evolution strategy with subjective selection is used to iteratively improve the appearance of the tone mapped images. Importantly, adjustments to the mapping are made in the interactive evolutionary process without a need for the user to understand the influence of the operators' parameters. A user of the system simply needs to pick the most appealing out of a set of automatically generated images. The motivation for blending the images obtained from several tone mapping operators is that mappings that cannot be generated by any one of the individual operators may be achieved. Rather than having to pick the operator most appropriate for the image at hand (and having to accept its limitations), a good mapping that may be outside of the range of any one operator can be found. We will also introduce an approach to screening the offspring of a generation such that the images presented for subjective selection will be constrained within a range of perceptual similarity values using a well known image metric.

The remainder of this thesis is organised as follows. Chapter 2 introduces the colour spaces, image metrics and tone mapping operators used in the present work as well as surveying relevant work on interactive evolutionary algorithms. Chapter 3 proposes an evolution strategy with subjective evaluation to solve the tone mapping problem. It will conclude with an informal evaluation of the proposed system. Lastly, Chapter 4 concludes with a brief summary and suggestions for future research.

A preliminary version [5] of this work has appeared in ACM SIGEVO's 11th

Annual Conference on Genetic and Evolutionary Computation.

## Chapter 2

### Background

#### 2.1 Overview

This chapter will provide an overview of the various areas of research related to topics found throughout this thesis. First we discuss the details of high dynamic range image formats and how they differ from the traditional image formats. The next section will detail the different colour spaces, which will be used later on. Thirdly we will discuss several of the image metrics, providing an understanding of their limitations and strengths, which will help to motivate the decision of which metric to use for our purposes. Then three tone mapping operators will be described, detailing their functions and algorithms. This chapter will conclude with an overview of some areas of evolutionary computation, looking at previous work on interactive evolution and constraint handling techniques.

#### 2.2 High Dynamic Range Image Formats

Traditionally images are stored in a 8-bits/channel or 24-bits/pixel format. As we saw in the introduction, high dynamic range images are those which capture a large range of luminance information. This amount of information is usually more than previous image formats are able to support. According to Reinhard et al. [34] there are three established HDR formats. These include the HDR, TIFF and EXR formats, each of which supports different types of pixel encodings. The encodings describe how the data is stored in the file and, depending on the type of encoding, some support compression. Compression is a desirable feature with many formats supporting 32-bits/channel or 96-bits/pixel. This can lead to some very large files if compression is not used, which for some applications such as the web can be a major drawback.

One interesting approach to high dynamic range formats is the one taken by Ward

and Simmons [48] with their JPEG sub-band encoding method. This method stores a tone mapped version of the image in the JPEG file as well as pixel data for the high dynamic range information. The HDR information is stored in a sub-band which is ignored by tools used to read and display the original JPEG files, this makes the HDR JPEG format backwards compatible. The HDR information can be used to reconstruct the image when the full HDR image is required. This format, just like the original JPEG format, is lossy in that the data is compressed beyond the point where all of the original can be recovered.

More information on the formats, encodings, compression, etc. can be found in [34] and [7]. The three established formats mentioned above all have freely available libraries which allow for the reading and writing of their files. Both the EXR and HDR formats are available through open source software through OpenEXR<sup>1</sup> and Radiance<sup>2</sup>, respectively.

### 2.3 Colour Spaces

Colour is the result of light entering the eye at different wavelengths. Light can either be reflected from objects or emitted from objects directly. The dominant wavelength of the light determines the hue of colour which we see. Visible light is roughly limited between 380 and 780 nm. Above this range is ultra violet and below infrared light, which are not visible to the naked eye.

Figure 2.1 shows the colour matching functions for CIELAB. The three curves  $\bar{x}(\lambda)$ ,  $\bar{y}(\lambda)$  and  $\bar{z}(\lambda)$  represent the auxiliary equations used to calculate the needed intensities the channels X, Y and Z, respectively, for a given colour. The Y primary for CIELAB was chosen such that the colour matching function  $\bar{y}(\lambda)$  matches the luminous-efficiency function. This allows for the channel Y to represent luminance, which allows us to quickly and easily determine luminance in this space. These colour matching functions are described for a viewing angle of  $2^\circ$ , which is important to note here as the perception of colour can change with different viewing angles. There are also colour matching functions described for a viewing angle of  $10^\circ$ , however this is

---

<sup>1</sup>OpenEXR: <http://www.openexr.com>

<sup>2</sup>Radiance: <http://www.radiance-online.org>

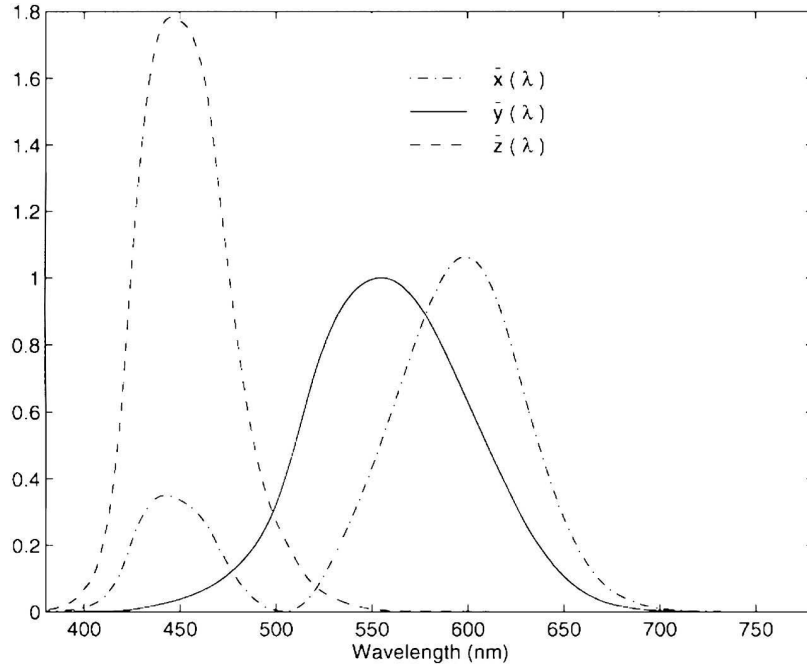


Figure 2.1: This plot from [39] shows the colour matching functions for 2° CIEXYZ.

not commonly used with computer graphics as this angle describes a much larger area of colour than what is typically found in computer graphics. Further details on colour as well as the details described here can be found in [15].

These colour matching functions were first described by the experiments of Stiles and Burch [41]. Using four light sources, three carefully selected primaries and one target, they experimented with human subjects asking them to tune the intensities of the three light sources till their combination matched the target light source [16]. With this data they were able to create three functions which give the needed intensities of the three primaries in order to produce the colour given a wavelength for each participant. The functions however varied slightly for each individual in the experiment, though not by much. Using this data the International Commission on Illumination (CIE) was able to come up with a standard set of equations for the XYZ colour space. Most colour spaces can be described in terms of their transformation from XYZ.

Another important topic in colour spaces is chromatic coordinates. Chromatic

coordinates describe the chroma information of a colour without the luminance information. The XYZ colour space has a corresponding set of chromaticity coordinates described by

$$\begin{aligned} x &= \frac{X}{X+Y+Z} \\ y &= \frac{Y}{X+Y+Z} \\ z &= \frac{Z}{X+Y+Z} = 1 - x - y. \end{aligned} \tag{2.1}$$

Since given  $x$  and  $y$  we are able to recover  $z$  the chromatic coordinates  $x$  and  $y$  define a 2D space. Many colour spaces are described by a single channel for luminance information and two separate channels for chroma information. For example, in the perceptually uniform colour spaces  $L^*a^*b^*$  and  $L^*u^*v^*$  channels  $a^*$  and  $b^*$  and  $u^*$  and  $v^*$  describe the chroma information for the two colour spaces respectively.

Several colour spaces are used throughout this work each chosen specific to the application for which it is used. The four that will be introduced in this section are RGB, XYZ,  $L\alpha\beta$  and  $L^*a^*b^*$ . Each of these play important roles in the calculation of different image attributes and image transformations. Further details may be found in [34].

### 2.3.1 RGB and XYZ

The RGB colour space is one of the most commonly used. It is seen in many image processing tools as the primary means for selecting colour. There are several different RGB colour spaces, however they all share the characteristic that they describe colour by specifying three intensities for red, green and blue. RGB is a device dependent colour space in that pixel values represented in RGB are tailored for a specific device depending on the device's white point and primary red, green and blue values. The white point describes the brightest white which the display is capable of producing. This is encoded by a tristimulus value in XYZ. This information is required when transforming an RGB image to the device independent colour space, XYZ. It is often used as a step in transforming images from RGB to another space due to this quality. Since RGB is device specific, transforming from RGB to other colour spaces can be difficult if the information about the device the image was targeted for is not available.



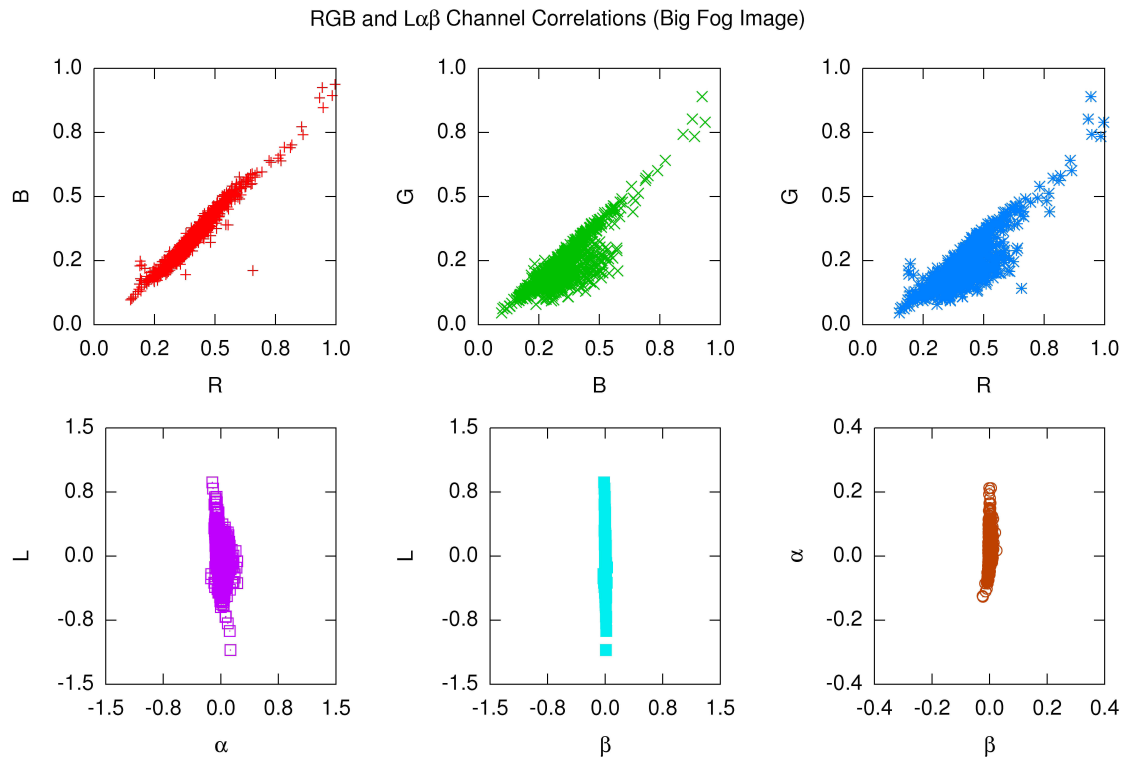


Figure 2.2: This example shows how the RGB colour channels are correlated and how  $L\alpha\beta$  is able to remove this correlation. In the first row of plots we can see that for the most part for a given contribution of one colour channel a very similar contribution of the other channels can be predicted. Looking at the second row of plots, we see that  $L\alpha\beta$  has transformed the data such that it is aligned with and roughly symmetric about the axes. This non-correlated colour space is important in that three dimensional image processing problems can be looked at as three separate one dimensional problems.

Reinhard et al. [34] give us that the second best solution is to use a standard matrix. One such matrix is provided by the International Telecommunications Union. Using this matrix the transformation from RGB to XYZ is:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.4306 & 0.3415 & 0.1784 \\ 0.2220 & 0.7067 & 0.0713 \\ 0.0202 & 0.1295 & 0.9394 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}. \quad (2.2)$$

### 2.3.2 Decorrelated Colour Space $L\alpha\beta$

The opponent colour space  $L\alpha\beta$  is less correlated colour space. The transformation to this colour space removes the much of the correlation which is present in RGB. The three components which make up the  $L\alpha\beta$  space are the single luminance channel  $L$  and the two chromatic channels  $\alpha$  and  $\beta$ . It is a common practice in many colour spaces to split the colour information in this way. Figure 2.2 shows how the RGB colour space is correlated by plotting each of the colour channels for the big fog image seen in Figure 3.15 against one another. The second row of plots in Figure 2.2 shows the  $L\alpha\beta$  colour channels of the same image plotted against each other. From these we see that axes have essentially been rotated and the data transformed such that the points now lie along the axes as well as now are symmetrical about the axes. This transformation is best described by Reinhard et al. in [31] which says that using several images, which are on average similar to many found in nature, Ruderman et al. [35] performed principal component analysis (PCA) on the RGB data of these images to determine the transformation which would give three decorrelated channels. These transformations for each image were then averaged and rounded to give the scale and rotation found in the transformation of Equation (2.4) below.

Having an uncorrelated representation of an image offers the advantage of being able to calculate image attributes in each of the channels separately rather than using more complex equations to consider all three correlated channels at once. This feature is exploited Reinhard et al. in [31] and by Toet and Lucassen [43] to create a similarity metric for colour images, which we will see later on in Section 2.4.3.

In order to transform an image to the  $L\alpha\beta$  colour space we first transform the device independent XYZ image to the LMS colour space. The LMS colour space separates the image into long, medium and short wavelengths. These correspond somewhat to the cones in the eye which also are sensitive to either long, medium or short wavelengths. This transformation is done by:

$$\begin{bmatrix} L \\ M \\ S \end{bmatrix} = \begin{bmatrix} 0.3897 & 0.6890 & -0.0787 \\ -0.2289 & 1.1834 & 0.0464 \\ 0.0000 & 0.0000 & 1.0000 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}. \quad (2.3)$$

After this transformation the log of each of the channels is computed in order to

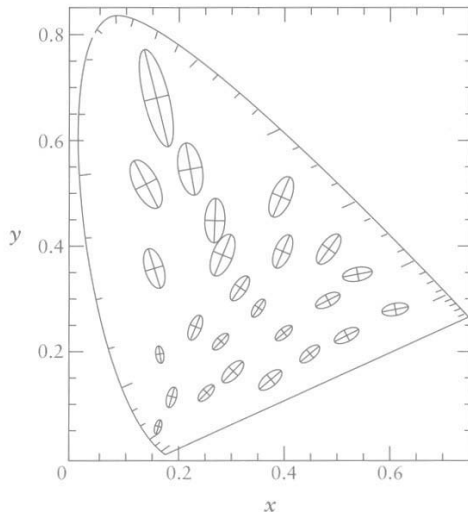


Figure 2.3: A plot of MacAdam ellipses used to show the non uniformity of XYZ taken from [16]. Each ellipse shows a region of colours which are indistinguishable from one another. These ellipses are enlarged so that they are easier to see. Their overall size is not important here, it is that their size and orientation relative to one another vary over the plot.

account for a skewness in the distribution of the intensities found in each channel. After which the final transformation to  $L\alpha\beta$  is performed by the following equation:

$$\begin{bmatrix} L \\ \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} 0.5774 & 0.5774 & 0.5774 \\ 0.4083 & 0.4083 & -0.8165 \\ 0.7071 & -0.7071 & 0.0000 \end{bmatrix} \begin{bmatrix} \log(L) \\ \log(M) \\ \log(S) \end{bmatrix}. \quad (2.4)$$

### 2.3.3 Perceptually Uniform Colour Space $L^*a^*b^*$

$L^*a^*b^*$ , also known as CIELAB, is a perceptually uniform colour space. This is a space in which the euclidean distance between colours is roughly proportional to the distance which is perceptually seen by a human observer. Glassner [16] describes perceptually uniform colour spaces as those in which interpolation from one colour to the next should behave “as expected”. For instance with the interpolation from  $\mathbf{c}_0$  to  $\mathbf{c}_1$  with

$$\mathbf{c} = (1 - a)\mathbf{c}_0 + a\mathbf{c}_1, \quad (2.5)$$

one might expect that the interpolation of colour should vary from  $\mathbf{c}_0$  to  $\mathbf{c}_1$  with respect to  $a$ . If using equal steps in  $a$  from 0 to 1 we would expect that the colour would also vary in equal steps. This is not true however in a non-perceptually uniform space such as XYZ. Figure 2.3 shows a plot of the XYZ space in the 2D chromaticity coordinates. The ellipses in Figure 2.3 show enlarged regions for which colour is perceptually constant. We notice that the ellipses are not the same size or orientation. This makes it rather difficult to define the relation of distances between the colour space to what is perceived. The CIELAB colour space attempts to transform the colour representation such that all of the ellipses are the same size and orientation. The results are not perfect, however. Figure 2.4 shows the MacAdam ellipses for the  $L^*u^*v^*$  colour space, a space similar to CIELAB. These spaces are both similar in that they both provide similar accuracy as a perceptually uniform colour space. They also both use a very similar set of functions to make the transformation from XYZ. In this figure we can see that the size and orientation of the ellipses have improved. There are other colour spaces which have been proposed which come close to a near perfect perceptually uniform colour space however their advantages are outweighed by the computational costs to transform an image. For this reason CIELAB and CIELUV are commonly used perceptually uniform colour spaces.

The distance measure in CIELAB is denoted  $\Delta E_{ab}^*$  which is evaluated as

$$\Delta E_{ab}^* = \sqrt{(\Delta L^*)^2 + (\Delta a^*)^2 + (\Delta b^*)^2} \quad (2.6)$$

which is the Euclidean distance between colour values. This gives us an intuitive description of the distances between colours and also allows for easy interpolation between colours.

The colour transformation from XYZ to CIELAB requires the knowledge of the XYZ colour values for the white illuminant of the scene, this is denoted as  $(X_n, Y_n, Z_n)$ . If the information regarding the white point is unknown, which is often the case, a standard illuminant is used. Using this illuminant we can calculate the  $L^*a^*b^*$  colour values as:

$$\begin{aligned} L^* &= 116f\left(\frac{Y}{Y_n}\right) - 16 \\ a^* &= 500 \left( f\left(\frac{X}{X_n}\right) - f\left(\frac{Y}{Y_n}\right) \right) \\ b^* &= 200 \left( f\left(\frac{Y}{Y_n}\right) - f\left(\frac{Z}{Z_n}\right) \right) \end{aligned} \quad (2.7)$$

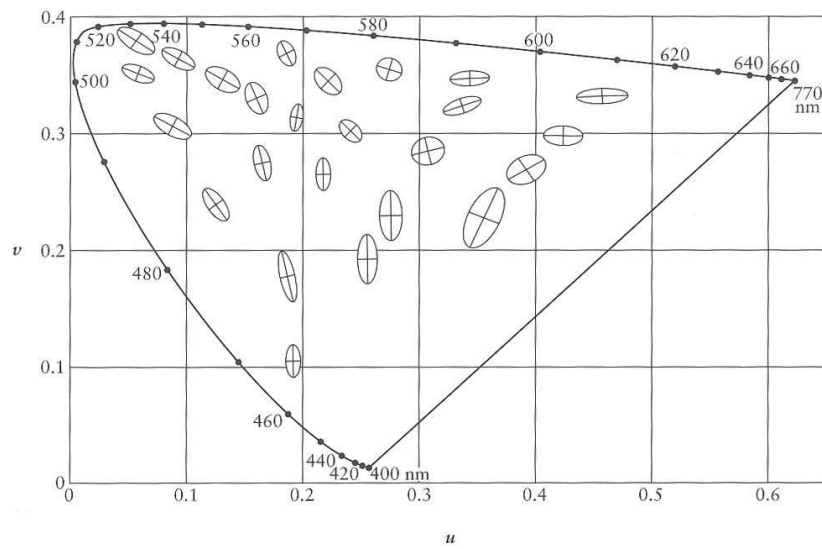


Figure 2.4: MacAdam ellipses for the CIELUV colour space taken from [16]. This colour space is more perceptually uniform. It is very similar to the CIELAB colour space and the reason that they both exist according to [34] is largely historic. CIELAB is commonly used in print material where CIELUV is more common with television. In this plot we can see that the ellipses are much closer in size and orientation when compared with those in Figure 2.3.

where the function  $f()$  takes the ratio of the of the tristimulus value against the tristimulus of the white point. This is how  $f()$  is defined:

$$f(r) = \begin{cases} (r)^{1/3} & : r > 0.008856 \\ 7.787r + \frac{16}{116} & : r \leq 0.008856 \end{cases} . \quad (2.8)$$

## 2.4 Image Metrics

Image metrics are most commonly used to test compression algorithms, to determine the amount of signal loss which occurs. Many of the first image metrics, such as the Mean Squared Error and the Peak Signal to Noise Ratio, come from other areas of signal processing. These metrics were designed to detect differences between signals. However, when it comes to the application of comparing images there are some types of changes which can be made to a signal which will be virtually undetectable to the viewer.

Recently there have been several new approaches to image quality assessment which account for the workings of the human visual system. For example, there is the very complex iCAM colour appearance model of Fairchild and Johnson [12]. The iCAM colour appearance model is able to calculate values for many image attributes such as contrast, lightness, chroma and hue. The model has been used for image comparison [13] and as well for tone mapping [20]. Another recent image metric is from the work of Wang and Bovik [45]. They have introduced the Universal Image Quality Metric as well as a further extension called the SSIM or Structural Similarity Image Metric. As implied by the name SSIM, these metrics base their quality assessment on the structural qualities of the image. These work quite well especially with greyscale images as structural distortions are most noticeable compared to those which do not distort structure such as changes to luminance and contrast. Convincing examples of different types of distortions can be found in [46].

This section will describe some of the image metrics in common use. The section will progress from a simple greyscale metric to a much more complex metric which accounts for the workings of the human visual system as well as colour images.

### 2.4.1 Mean Squared Error and Peak Signal to Noise

Mean Squared Error (MSE) is commonly used in many signal processing applications to compare two signals. The metric has been in use for over 50 years according to Wang and Bovik [46] who compare MSE against some of the more recent signal fidelity measures. The process for calculating this metric is quite straightforward. Given two discrete signals  $\mathbf{x}$  and  $\mathbf{y}$  of length  $N$ , we compare these by

$$\text{MSE}(\mathbf{x}, \mathbf{y}) = \frac{1}{N} \sum_{i=1}^N (x_i - y_i)^2. \quad (2.9)$$

In the application of image processing the signals  $\mathbf{x}$  and  $\mathbf{y}$  represent the pixel intensities and  $N$ , the number of pixels in each image.

According to Wang and Bovik a few of the reasons why MSE is commonly used are its relatively simple calculation with a low computational cost and it exhibits several advantageous mathematical properties. Some of these properties include non-negativity, symmetry, a single identity and that it exhibits the triangular inequality property.

Another metric commonly used in measuring signal quality is the peak signal fidelity or peak signal to noise ratio (PSNR) [46]. It is calculated using MSE and remains very similar, however it takes into account the dynamic range of each signal. This is useful when comparing two signals with different dynamic ranges. PSNR can be calculated as

$$\text{PSNR}(\mathbf{x}, \mathbf{y}) = 10 \log_{10} \frac{L^2}{\text{MSE}(\mathbf{x}, \mathbf{y})} \quad (2.10)$$

where  $L$  is the dynamic range of the signal, which for a greyscale 8 bit low dynamic range image is equal to 255.

### 2.4.2 Universal Quality Index

The universal quality index was first introduced by Wang and Bovik in [45]. The main issue with MSE and PSNR is that they do not take into account how the human visual system (HVS) works. As a result of this they do not correlate well with human perception.

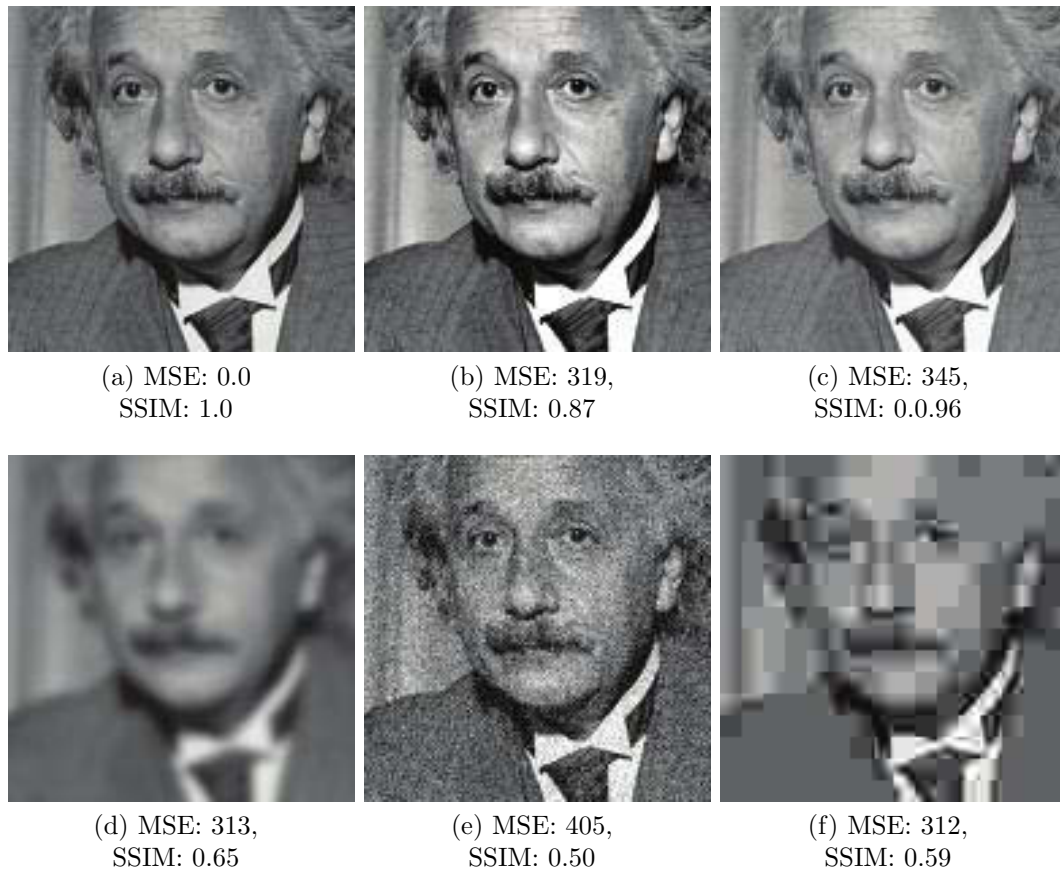


Figure 2.5: This figure compares Mean Squared Error (MSE) with the Structural Similarity Image Metric (SSIM) for different types of distortions. These images come from a similar figure in [46], these images have been distorted such that each have a similar MSE score and this leaves open for interpretation the scores of the SSIM. The distortions are: (a) Original, (b) Contrast Stretch, (c) Luminance Shift, (d) Blurring, (e) Gaussian Noise, (f) JPEG Compression. Here it should also be noted that SSIM has a maximum of 1.0 obtained when the two images being compared are identical. MSE on the other hand has an optimal value of 0.0 as it is a measure of error.



Many image quality methods are intended to measure the impact that noise, compression and other factors have on the quality of the image signal. Wang and Bovik give some examples in [45] where MSE fails, showing three images with a similar MSE score where one image appears virtually unchanged from the original and the others have very obvious signal degradations and distortions. They show that their quality index is able to give a high score to the image which appears most unchanged and much lower scores to the others. We have recreated a similar series of examples in Figure 2.5.

The approach taken by Wang and Bovik is based on the assumption that the HVS is most sensitive to detecting distortion in the structural information of an image. The structural information in a scene is made up of common objects such as facial features which the HVS is highly adapted to. The image metric makes its full comparison using structural, luminance and contrast information contained in the images.

We will start with the details for how the luminance and contrast signals are compared. For two greyscale image signals  $\mathbf{x}$  and  $\mathbf{y}$ , luminance is measured by the mean pixel intensities. This comparison is computed by the equation

$$l(\mathbf{x}, \mathbf{y}) = \frac{2\bar{x}\bar{y}}{(\bar{x})^2 + (\bar{y})^2} \quad (2.11)$$

where  $\bar{x}$  and  $\bar{y}$  are the means for the image signals  $\mathbf{x}$  and  $\mathbf{y}$  respectively.

Contrast in each signal is measured using the standard deviation of each signal denoted  $\sigma_{\mathbf{x}}$  and  $\sigma_{\mathbf{y}}$  for signals  $\mathbf{x}$  and  $\mathbf{y}$  respectively. The standard deviations are compared the same way as the means in the luminance comparison. This is done with the equation:

$$c(\mathbf{x}, \mathbf{y}) = \frac{2\sigma_{\mathbf{x}}\sigma_{\mathbf{y}}}{\sigma_{\mathbf{x}}^2 + \sigma_{\mathbf{y}}^2}. \quad (2.12)$$

Lastly we will describe how the structure of the image signals is measured and compared. Wang and Bovik describe structure using a projection of the image signal onto the hyperplane described by

$$\sum_{i=1}^N x_i = 0. \quad (2.13)$$

This projection is done only after the each image signal is normalized. The normalization process is calculated by subtracting from each image signal its mean intensity and then dividing the signal by its standard deviation. The correlation between the normalized signals,  $(\mathbf{x} - \bar{x})/\sigma_x$  and  $(\mathbf{y} - \bar{y})/\sigma_y$ , give the measure for structural similarity. The structure comparison is defined as:

$$s(\mathbf{x}, \mathbf{y}) = \frac{\sigma_{\mathbf{xy}}}{\sigma_{\mathbf{x}}\sigma_{\mathbf{y}}} \quad (2.14)$$

where  $\sigma_{\mathbf{xy}}$  is computed as

$$\sigma_{\mathbf{xy}} = \frac{1}{N-1} \sum_{i=1}^N (x_i - \bar{x})(y_i - \bar{y}). \quad (2.15)$$

The quality index then combines these three calculations by multiplying them to give:

$$Q(\mathbf{x}, \mathbf{y}) = \frac{\sigma_{\mathbf{xy}}}{\sigma_{\mathbf{x}}\sigma_{\mathbf{y}}} \frac{2\bar{x}\bar{y}}{(\bar{x})^2 + (\bar{y})^2} \frac{2\sigma_{\mathbf{x}}\sigma_{\mathbf{y}}}{\sigma_{\mathbf{x}}^2 + \sigma_{\mathbf{y}}^2}. \quad (2.16)$$

This metric can become unstable when the standard deviations and mean intensities are close to zero, however this issue was later addressed by Wang and Bovik [47] in a metric which is known as the structural similarity image metric (SSIM). This addresses the issue of stability by introducing a few new constants into the above equations.

Since the HVS analyzes images locally rather than as a whole Wang and Bovik note that it is best to make the calculations using a small sliding window over the image. The local calculations can then be averaged to give a signal metric value. Different types of windows were tested in [47], where it was found that the results which provide the tightest correlation with that of human observers is an  $11 \times 11$  circular-symmetric Gaussian weighting function.

### 2.4.3 Universal Colour Image Fidelity Metric

The universal colour image fidelity metric was introduced by Toet and Lucassen in [43] in which they extend the universal image quality index to colour images. They state that by simply applying the greyscale metric to the three RGB colour channels we form a metric for colour images. However, this metric does not correlate well with what is perceived by humans.

Figure 2.2 shows how strongly correlated the RGB channels are, where contributions of the channels are usually matched by the others. The authors use this observation as their motivation for using a perceptually uncorrelated colour space. This allows us to analyze each channel individually. For the uncorrelated colour space, the space  $L\alpha\beta$ , described in Section 2.3.2 is used. With this the colour fidelity metric is defined as

$$Q_{colour}(\mathbf{x}, \mathbf{y}) = \sqrt{w_L(Q(\mathbf{x}_L, \mathbf{y}_L))^2 + w_\alpha(Q(\mathbf{x}_\alpha, \mathbf{y}_\alpha))^2 + w_\beta(Q(\mathbf{x}_\beta, \mathbf{y}_\beta))^2} \quad (2.17)$$

where  $Q(\mathbf{x}, \mathbf{y})$  is the image quality metric function described in the previous section and the subscripts  $L$ ,  $\alpha$  and  $\beta$  on the vectors  $\mathbf{x}$  and  $\mathbf{y}$  denote the image vectors of the single colour channel. Also the weights,  $w_L$ ,  $w_\alpha$  and  $w_\beta$  are present to give the results of the quality index for each colour channel different contributions to the overall colour fidelity metric.

Toet and Lucassen performed experiments where observers ranked images based on their degradation. Using this information they were able to show how well correlated their measure is with human perception. They were also able to determine the best set of weights for the three channels, for the two images that were experimented with, which gave the tightest correlations. Their findings for the weights pinned  $w_L$  with the largest of the three with values of 3.3 and 2.8. The other two weights,  $w_\alpha$  and  $w_\beta$  were both close to 1.0, with  $w_\alpha$  result with values of 1.3 and 0.9 and  $w_\beta$  with values of 0.9 and 0.8.

## 2.5 Tone Mapping Operators

Tone mapping high dynamic range images is the process of reducing the range of the luminance values such that they will fit into the range available on the target viewing device. The most common range for viewing devices is the 24bit RGB colour range where each colour channel is able to store 256 different values. Existing tone mapping operators can all be broadly categorized as being either global or local. Global operators perform mappings by fitting all of the real world intensities to a single curve whereas local operators will adapt the curve for each pixel in the image in order to maximise features locally. Each of these types of operators have their own

advantages and disadvantages. Local operators are best at preserving local contrast. However these operators are also slower to compute compared to global operators and can sometimes suffer from halo artifacts around areas where very bright regions border much darker ones. Global operators on the other hand are quite fast at tone mapping since they are not calculating values over neighbourhoods for each pixel, but they do suffer from some loss in local contrast as all of the real world intensities are fitted to a single curve.

Tone mapping operators can also be classified as to whether they follow traditional computer graphics methods of dynamic range reduction, using processes such as gamma corrected clamping, or whether they integrate research which has emerged from psycho-physical models of the human visual system (HVS). There are also operators which do not follow either models of the HVS or traditional CG methods. For example Fattal et al. propose a method that compresses the dynamic range of the image in the gradient domain [14].

Many tone mapping operators simply deal with compressing the luminance channel of an image [34]. It is generally accepted that the majority of the dynamic range in a high dynamic range scene is contained within the luminance channel, with less in the chromatic channels. However, it should also be noted that some of the more recent tone mapping operators take into account the mapping of colour in a more sophisticated way. The operators which map luminance will simply reduce the dynamic range of the luminance channel by applying the transformation (shown in [34]):

$$\begin{bmatrix} R^d \\ G^d \\ B^d \end{bmatrix} = \begin{bmatrix} L^d \frac{R^w}{L^w} \\ L^d \frac{G^w}{L^w} \\ L^d \frac{B^w}{L^w} \end{bmatrix}. \quad (2.18)$$

This reduces the world RGB values,  $(R^w, G^w, B^w)$ , to low dynamic range display values,  $(R^d, G^d, B^d)$ . This is a linear mapping with respect to the world luminance values  $(L^w)$  and their new calculated display values  $(L^d)$ . Luminance can be taken from the XYZ space as Y represents luminance as described in Section 2.3. With this we are able to calculate the luminance using the middle row of the matrix from Equation (2.2) which yields the calculation for luminance as:

$$L = 0.2651R + 0.6702G + 0.0641B. \quad (2.19)$$

The notation of superscript  $d$  and  $w$  will be used through this thesis to denote display and world quantities respectively. The operators used in our evolutionary tone mapping tool are those of Schlick [36] and Ashikhmin [1] as well as that of Reinhard and Devlin [32]. This selection includes diverse approaches in order to span as large of a mapping space as possible. Further considerations included the existence of parameters that can be tuned as well as computational speed. Both Schlick’s and Reinhard and Devlin’s operators are global; Ashikhmin’s is a local operator and the only one of the three to consume significant computational resources. While Schlick’s operator implements a simple heuristic, both Ashikhmin’s and Reinhard and Devlin’s are based on psycho-physical models of photoreceptors.

### 2.5.1 Schlick’s Operator

Schlick’s algorithm [36] is one of the earliest tone mapping operators. It attempts to improve on some of the more traditional computer graphics techniques for dynamic range compression. But before introducing Schlick’s operator it is useful to introduce a few other tone reproduction techniques. These techniques will use a similar structure, mapping luminance  $L^d(x, y)$  as

$$L^d(x, y) = \lfloor L_{max}^d F(L^w(x, y)) \rfloor \quad (2.20)$$

where  $F()$  is a mapping defined as:

$$F : [0, L_{max}^w] \longrightarrow [0, 1]. \quad (2.21)$$

The most primitive technique is a simple linear mapping. In this case  $F$  linearly maps the range  $[0, L_{max}^w]$  into  $[0, 1]$  by

$$F(L^w(x, y)) = L^w(x, y) / L_{max}^w. \quad (2.22)$$

The rest of this section will describe different variations on  $F$  which improve upon this. The values mapped to  $L_{max}^d$ , the maximum display luminance, will then be clamped to  $L_{max}^d - 1$  as values between 0 and  $L_{max}^d - 1$  are desired. It has been said [1] that linear mappings best preserve detail, however these mappings do not

take into account the non-linear response of most display devices. Gamma-corrected linear mappings can improve upon this. The mapping replaces  $F$  with

$$F^\gamma(L^w(x, y)) = \left( \frac{L^w(x, y)}{L_{max}^w} \right)^{1/\gamma} \quad (2.23)$$

where  $\gamma \in [1, 3]$ . Although this method does help to correct for the non-linear response of the display device, for some images it sacrifices detail in dark areas of the image. This issue however is addressed by the gamma-corrected clamping which is used to preserve some of this detail. With gamma-corrected clamping we change the function  $F$  to

$$F^{p,\gamma}(L^w(x, y)) = \begin{cases} (L^w(x, y)/p)^{1/\gamma} & : L^w(x, y) < p \\ 1 & : \text{otherwise} \end{cases} \quad (2.24)$$

where  $p \in [L_{min}^w, L_{max}^w]$  and  $\gamma \in [1, 3]$ . This method works well but as Schlick points out there is no easy way to decide on values for  $p$  and  $\gamma$ . The parameter  $\gamma$  is often given for display devices, however this can change depending the monitor's brightness and contrast settings as well as the level of light in the environment where the device is located. There is also a logarithmic mapping which uses

$$F^{p,\gamma}(L^w(x, y)) = \left( \frac{\log(1 + pL^w(x, y))}{\log(1 + pL_{max}^w)} \right)^{1/\gamma} \quad (2.25)$$

where  $p \in [0, \infty)$  and  $\gamma \in [1, 3]$ . This can produce good results however the logarithmic operation has a fairly high computational cost for some applications, and there remains the difficulty of choosing appropriate values for both  $p$  and  $\gamma$ . These are the motivations behind the tone mapping function proposed by Schlick.

The following function is a rational tone mapping function which has one parameter (which will be explained later) that is easily tuned

$$F^p(L^w(x, y)) = \frac{pL^w(x, y)}{pL^w(x, y) - L^w(x, y) + L_{max}^w} \quad (2.26)$$

where  $p \in [1, \infty)$ . By changing  $p$  this operator is able to provide curves similar to both logarithmic and exponential curves without the added cost of their computation. Finding suitable values for  $p$  may be difficult. However, Schlick addresses this issue with an equation for calculating an appropriate value for  $p$  based on an input parameter  $M$ . The parameter  $p$  is calculated as

$$p = \frac{ML_{max}^w}{L_{max}^d L_{min}^w} \quad (2.27)$$

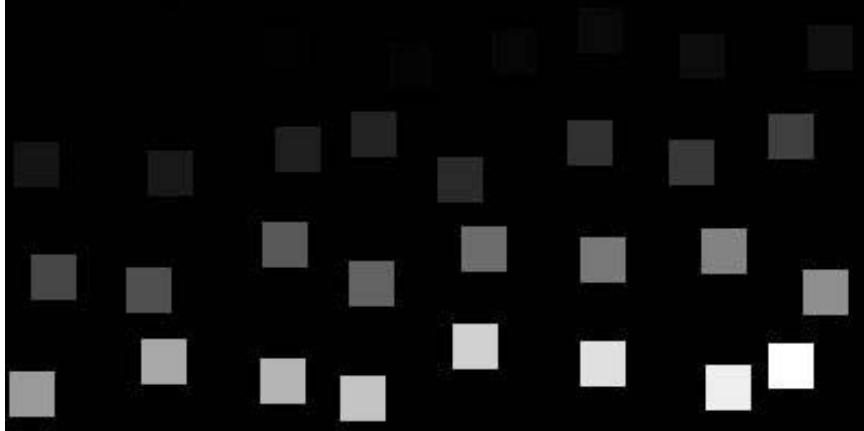


Figure 2.6: This image is provided by Schlick in [36] for calculating the darkest grey visible on a display device. By viewing this image on the target display device and selecting the first square in the image which is distinguishable from the black background, this gives the intensity which can be use as a value for the parameter  $M$ .

where  $M$  is the display value of the darkest grey distinguishable from black on the target device. They provide the image in Figure 2.6 which can be used to find this value. Figure 2.7 shows the different curve shapes for exponential and logarithmic mappings as well as the curves capable of Schlick's rational mapping for different values of the parameter  $M$ . As we can see the rational mappings are able to form curves quite similar to those found in the logarithmic and exponential mappings. Schlick's goal of removing the parameters has not been completely met. However the guesswork in finding good values for  $p$  and  $\gamma$  is removed, which is one of the major drawbacks of the other mappings introduced above. While simple, Schlick's operator is often surprisingly effective. Equation (2.26) defines a sigmoidal compression curve the exact shape of which is determined by parameter  $M$ . Small changes in  $M$  have a small effect on the shape of the curve, making it a valuable parameter that can be tuned in the evolutionary approach. This operator has just the one parameter to tune, the value for  $M$ .

### 2.5.2 Ashikhmin's Operator

Ashikhmin's tone mapping operator [1] is a local operator developed with the goal of finding a good balance between closely modelling the human visual system on the

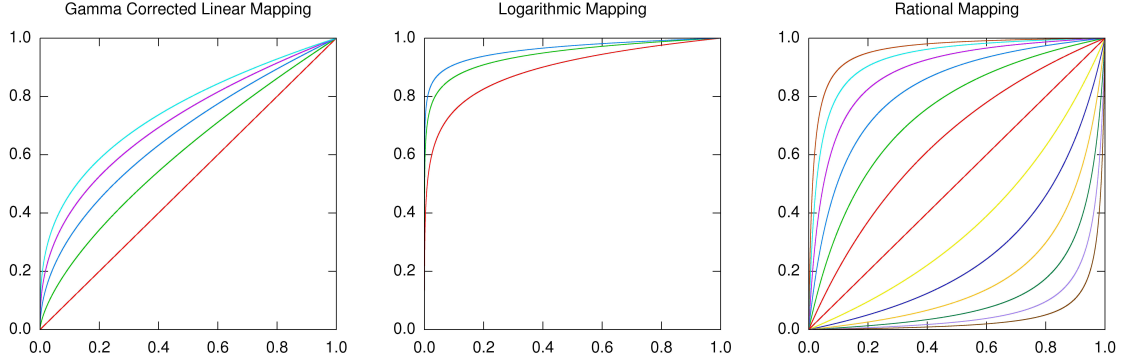


Figure 2.7: These plots show some curves representing gamma corrected linear mappings (or exponential), logarithmic mappings and rational mappings. As we can see the rational mappings are able to adopt shapes similar to those of the exponential and logarithmic mappings as well as form further shapes. When viewed over a logarithmic  $x$ -axis the rational mappings exhibit a sigmoidal shape. This allows for the extreme ends of the compression range to be treated differently than the rest.

one hand and simplicity on the other. With this operator the intention is to model two features of the HVS. These include absolute brightness and local contrast. This section will discuss the derivation of the operator and how it will achieve these two features.

Firstly, contrast is described as

$$c(x, y) = L^w(x, y)/L_a(x, y) - 1 \quad (2.28)$$

where  $L_a(x, y)$  is the local adaptation level or background intensity at the pixel location  $(x, y)$  in the scene. This describes contrast in relation to the surrounding intensities. In order to preserve local contrast the constraint  $c^d(x, y) = c(x, y)$  is used, which implies that the contrast present in the display image should be equal to the contrast of the original. Now consider display contrast to be of the form

$$c^d(x, y) = L^d(x, y)/\text{TM}(L_a(x, y)) - 1 \quad (2.29)$$

where  $\text{TM}()$  is the tone mapping function later described for the local adaptation level. This contrast is described in terms of display quantities. Combining Equations (2.28) and (2.29) with the constraint that  $c^d(x, y) = c(x, y)$  we then have a function for display luminance in terms of world luminance. This gives the relation:

$$L^d(x, y) = \frac{L^w(x, y)\text{TM}(L_a(x, y))}{L_a(x, y)}. \quad (2.30)$$



For most tone mapping applications visible contrast is more important to preserve than the one calculated with Equation (2.28). Ashikhmin describes visible contrast as

$$c_v(x, y) = (L^w(x, y) - L_a(x, y))/\text{TVI}(L_a(x, y)). \quad (2.31)$$

This equation uses the TVI function which is the threshold versus intensity function. This gives the value of the just noticeable difference (JND) for the given adaptation level. In other words, the TVI function gives the intensity of light which is just noticeable from a given background intensity. We can also describe visible contrast with display quantities as we did before with contrast. For this we have:

$$c_v^d = (L^d(x, y) - \text{TM}(L_a(x, y)))/\text{TVI}(\text{TM}(L_a(x, y))). \quad (2.32)$$

Now if we keep the constraint that contrast in the original image should be equal to that in the final image and apply this to visible contrast we then have the relation  $c_v^d = c_v$ . This leads us to the final equation for display luminance which is described by:

$$L^d(x, y) = \text{TM}(L_a(x, y)) + \frac{\text{TVI}(\text{TM}(L_a(x, y)))}{\text{TVI}(L_a(x, y))} (L^w(x, y) - L_a(x, y)). \quad (2.33)$$

This gives us a function which maps world luminance to display luminance while preserving visible contrast. What we have yet to describe is how the tone mapping function TM and the local adaption are defined. In order to preserve as much detail as possible the tone mapping function is calculated using a linear mapping. The tone mapping function will not linearly map luminance directly, rather it will use perceptual capacity for the mapping. The perceptual capacity is the number of visible JNDs for a given background intensity. This can be calculated by integrating the TVI function. However, the TVI function itself would be difficult to integrate analytically and so a close approximation is made in log-log space using linear segments. These equate to the piece-wise function

$$C(L) = \begin{cases} L/0.0014 & : L < 0.0034 \\ 2.4483 + \log(L/0.0034)/0.4027 & : 0.0034 \leq L < 1 \\ 16.5630 + (L - 1)/0.4027 & : 1 \leq L \leq 7.2444 \\ 32.0693 + \log(L/7.2444)/0.0556 & : \text{otherwise} \end{cases}. \quad (2.34)$$

The tone mapping function that linearly maps perceptual capacity to display luminance is

$$\text{TM}(L_a) = L_{max}^d \left( \frac{C(L_a) - C(L_{min}^w)}{C(L_{max}^w) - C(L_{min}^w)} \right). \quad (2.35)$$

Lastly we need to describe how local adaptation is calculated. First  $s_{max}$  different images are created. These are denoted  $G_s(L)$  which are the results of applying a Gaussian filter of width  $s$  to the image. Local adaptation is taken to be  $G_s(L)$  for the largest  $s$  for which local contrast is not greater than a threshold or  $s$  no larger than  $s_{max}$ . Ashikhmin reports ten as a reasonable default value for  $s_{max}$  as it was found to be adequate.

Local contrast is calculated as

$$lc(s, x, y) = (G_s(L)(x, y) - G_{2s}(L)(x, y)) / G_s(L)(x, y). \quad (2.36)$$

Determining the Gaussian neighbourhood size based on local contrast is how this operator is able to avoid halo artifacts. It should be noted though that halos do still appear, but by tweaking the parameters to suit the image it is possible to control the degree to which they do occur. The parameters that this operator has for adjusting results are the local contrast threshold and a scale factor. The scale factor is used in a preprocessing stage to, if necessary, scale the units of the image so that they are transformed to Standard International (SI) units. This step was noted as being necessary in [34]. Transforming the image to SI units ensures that the luminance values are described in terms of candelas per square metre which is the unit of measurement for luminance required for the TVI function.

### 2.5.3 Reinhard and Devlin

The operator of Reinhard and Devlin [32] also models properties of the human visual system. The model includes several parameters which control intensity, contrast, lightness, and chromatic adjustment. Underlying the operator is the insight that the photoreceptor potential to distinguish luminance differences is relative to the current intensity it is experiencing as well as recent intensities. The operator is described by

$$I_{r|g|b}^d(x, y) = \frac{I_{r|g|b}(x, y)}{I_{r|g|b}(x, y) + (fA_{r|g|b}(x, y))^m} \quad (2.37)$$

where  $I_{r|g|b}(x, y)$  are the intensities of the current pixel in the red, green and blue channels of the input image and  $I_{r|g|b}^d(x, y)$  are the corresponding intensities in the output image. Parameter  $f$  provides control over the overall intensity of the output image while  $m$  determines the shape of the compression curve and allows one to trade contrast in medium intensity regions of the image for detail in the dark and bright regions. The adaptation level  $A_{r|g|b}(x, y)$  is computed as

$$A_{r|g|b}(x, y) = aA_{r|g|b}^{local}(x, y) + (1 - a)A_{r|g|b}^{global} \quad (2.38)$$

where  $A_{r|g|b}^{local}(x, y)$  and  $A_{r|g|b}^{global}$  denote local and global adaptation levels, respectively, and parameter  $a$  determines the relative weighting of the two. The local adaptation level is computed as

$$A_{r|g|b}^{local}(x, y) = cI_{r|g|b}(x, y) + (1 - c)L(x, y) \quad (2.39)$$

where  $L(x, y)$  is the pixel's luminance. The global adaptation level

$$A_{r|g|b}^{global} = cI_{r|g|b}^{average} + (1 - c)L^{average} \quad (2.40)$$

is computed from intensity and luminance values averaged across the image. Weighting factor  $c$  allows for colour correction in that for values close to one, each colour channel is treated independently whereas for values close to zero, overall luminance dominates.

Reinhard and Devlin [32] point out that many operators offer several parameters to tune. However usually these control unintuitive properties with very large and perhaps unbounded ranges. As mentioned there are four properties for which parameters are provided. Parameter  $a$  gives control over the correlation of the color channels and  $c$  gives the interpolation between global and local intensity adaptation. Parameters  $m$  and  $f$  give control over contrast and intensity, respectively. For the parameter  $m$  a method for automatically calculating a 'good' setting is given. This method can be used if the user does not provide their own setting. This automatic calculation is done by:

$$m = 0.3 + 0.7k^{1.4} \quad (2.41)$$

where  $k = (L_{max}^w - L_{avg}^w)/(L_{max}^w - L_{min}^w)$ .

The three tone mapping operators described here together offer several parameters to tune, a gamma encoding is also used as part of the post processing stage. These parameters will all be valuable features to include as dimensions of our interactive evolutionary search space.

## 2.6 Interactive Evolution

Evolutionary strategies employ techniques found in natural evolution to solve optimization problems. At each generation offspring are generated as a combination of random mutations of one or more of the parents. Once the population is complete for a generation all of the candidate solutions are evaluated against a fitness function. The results of the fitness function evaluation are used to determine which candidate solution or solutions will be the parent or parents of the next generation. This process is repeated until certain conditions are met. These conditions can include converging on an optimal fitness value, a maximum number of generations, a maximum length of time passing, etc.. Often, these strategies use step length and covariance matrix adaptation which can help to increase the speed at which the algorithm converges to the goal. Step length here refers to a variable which is used to control the strength of the mutations. Covariance matrix adaptation is used to find dependencies in the dimensions of the search space and uses this insight to shape the mutations accordingly. A detailed introduction to evolutionary strategies can be found in [3].

Interactive evolutionary algorithms are iterative optimisation strategies that are applied to problems for which the objective is not easily formalised but instead is judged subjectively. One of the most impressive results generated so far is the interactive evolution of coffee blends by Herdy [19]. A discussion of the approach, as well as difficulties, application areas, and perspectives are discussed by Banzhaf [2]. A more recent survey of a large number of applications has been compiled by Takagi [42]. In fact in this 2001 survey Takagi reports 251 publications where interactive evolutionary computation is being used spanning many different applications.

User fatigue has repeatedly been recognised as a significant obstacle in interactive evolutionary algorithms. It is typically expected that a human subject can or is willing to provide meaningful input for a maximum of between ten and twenty generations.

Moreover, it is sometimes desirable to reduce strain on the user by asking him or her to perform simple selection tasks rather than more difficult ranking tasks. A thorough discussion of user interaction strategies can be found in Breukelaar et al. [4]. Takagi [42] asserts that the ability for users to remember the different offspring when viewing them sequentially becomes an issue especially for time varying data such as videos or images with too high a level of detail to be displayed simultaneously. It is therefore necessary to design a solution which allows the users to view and compare the offspring with minimal effort.

As a result of the typically small number of iterations that interactive evolutionary algorithms are limited to, modern approaches to step length or even covariance matrix adaptation that may require many time steps before generating useful results are impractical. Nonetheless, choosing appropriate step lengths is essential as too small steps require too many generations in order to yield useful results, while too large steps quickly lead to user frustration. Experiments involving interactive evolution strategies with successful step length adaptation have been reported by Herdy [18]. This work also emphasises the benefits of recombination for interactive evolution. An interesting approach to circumventing the problem of user fatigue is the recent approach by Picbreeder [37, 38], a computational online tool for evolving images that allows users to pick up improving candidate images where others have left off. This obviously can only work depending on the application. For instance for something requiring a more immediate solution, depending on users to voluntarily mutate the solutions via a website may not be practical.

## 2.7 Constraint Handling

In evolutionary computation, constraint handling is used when limitations are set on the dimensions of the search space. Several methods have been introduced in the past which are described in a survey by Michalewicz [26]. Many constraint handling techniques use a penalty function which determines how the evolutionary model deals with offspring that exceed the constraints. The use of the different techniques can depend on the nature of the fitness function and the topology of the search space. This section will describe a few of the methods detailed in the Michalewicz survey.

We restrict this discussion to box constraints as this is the only type of constraint found in this thesis.

Some of the most straightforward approaches to constraint handling include rejecting and repairing offspring. With rejecting offspring, any offspring which does not lie within the constraints is not considered as a candidate. Repairing offspring on the other hand will adjust the offspring so that they will all lie within the constraints. This can be done by clamping the individual dimensions to the constraints when necessary or by interpolating between the parent and the offspring vectors until all the parameters are inside the constraints. These constraint handling methods will only produce offspring which lie in the search space. However, this may not be desirable in all circumstances as it has been noted that if the search space is not contiguous, this strategy may make crossing regions through the space where non-feasible solutions lie difficult or impossible.

Lastly there are constraint handling techniques which allow for both feasible and non-feasible solutions to be considered as a solution. These methods are usually much more complex than those described above. Usually this is done by incorporating a factor into the fitness function to account for how much a non-feasible solution exceeds the constraints. These types of methods are described in [30] and [27].

## Chapter 3

### Interactive Evolutionary Tone Mapping

We propose interactive evolution as a computational tool for tone mapping. Using subjective selection the tool will provide users with a simple process for tone mapping high dynamic range images. The tool will allow the user to select an appropriate operator along with parameter settings for the image which they are tone mapping. This will be done without requiring knowledge of the operators and parameters by the user. Instead the operator or operators and their parameters are selected through an evolutionary process.

The proposed solution is a  $(1 + \lambda)$ -ES with subjective selection. This chapter will describe the details of our interactive evolutionary tone mapping tool. The approach includes a weighted blending of images which are the results of multiple tone mapping operators. The weights allow for the control of each operator's contribution to the tone mapped image. This removes the problem of deciding which operator is best suited for the image as the evolutionary strategy will be able to mutate towards a good mapping deciding on the best operator or a combination thereof. The strategy may also be able to find a solution outside the range of each individual operator, creating the potential for results which a single operator may not be capable of. The weights assigned to each operator controlling their contributions make it possible to eliminate operators which are ill suited for the particular image as their weights can mutate towards zero. As it was pointed out in [32] and [36], it is often difficult to assign appropriate values for the parameters of each operator, and so these will be mutated as well.

The operators which will be selected for this task must be fast since each offspring requires its own run of each algorithm. Also tunable parameters are an attractive feature for this task as they add dimensions to our search space. The operators chosen for this thesis are the ones discussed in Section 2.5, these are the operators of

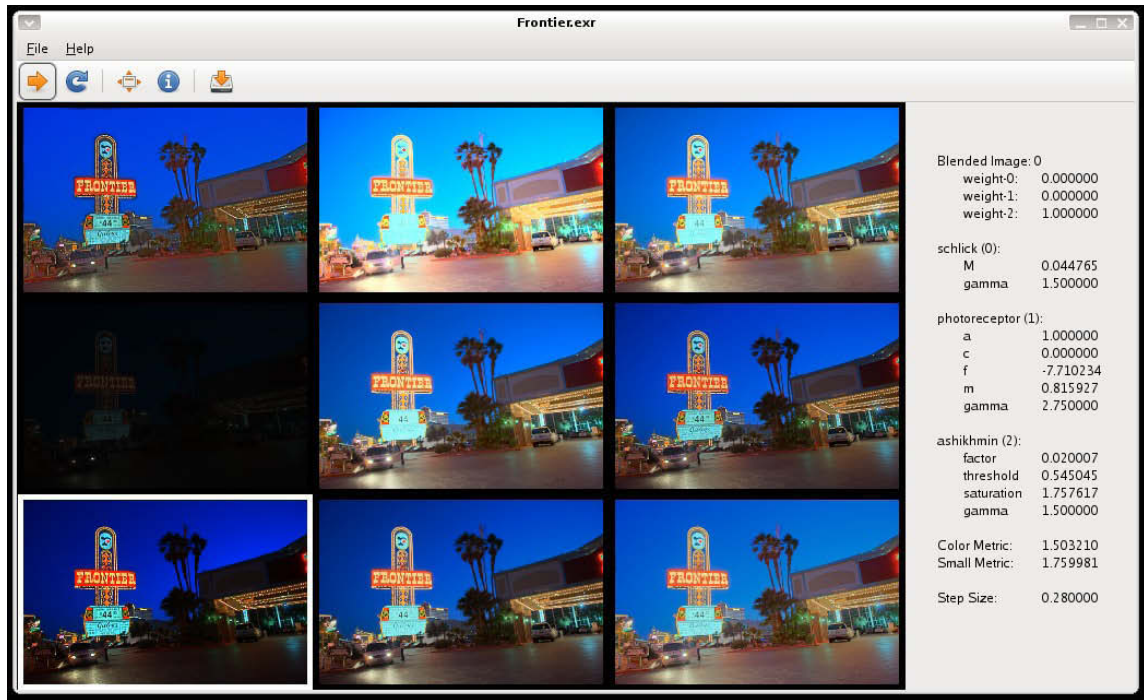


Figure 3.1: A screen shot of the interface. The user is presented with nine images with a diverse selection. There are a few which are beyond anything that might be considered desirable. However, there are several very eligible candidates each with unique characteristics.

Ashikhmin, Schlick and Reinhard and Devlin. Clearly, other or more operators could easily be accommodated. The implementation of the operators as well as several image processing routines, provided by Reinhard et al. [34], have been used as the basis of our program.

The chapter will first introduce the basic algorithm. This will detail the user interface, the blending of the images, the generation of the offspring and how step size is controlled. After which we will provide the details of another method for controlling the diversity of the offspring. This will use the image metric introduced in Section 2.4.3. This metric will allow the offspring to be filtered based on perceptual similarity before the generation is presented to the user. The chapter will conclude with a discussion of the tool and results.



### 3.1 Basic Algorithm

The user interface of our evolutionary tone mapping tool as shown in Figure 3.1 is deliberately simple. The goal is to provide a clean and easy to use interface which does not overwhelm the user with too much information. This allows the user to easily and quickly make a decision. After some experimentation, we have set the number of offspring,  $\lambda$ , to 8, as providing fewer images sometimes provided too little variability, while displaying more images required extra effort from the user to make a selection. The images are displayed to the user in equal size, with the parent always appearing in the same location. The user can select an image by double clicking and enlarge it if desired. There is also a feature which enables the user to start over in case they feel that none of the displayed images will lead to a good solution. On a recent PC with a quad core processor, generating a new generation of tone mapped images takes about ten seconds. This of course depends on the size of the images. This time could be reduced by only using global operators since they are generally faster to compute. However, we wanted to incorporate the use of a local operator as it gives us more variety in our mapping algorithms.

#### 3.1.1 Search Space

With the operators we have chosen, we have a total of seven input parameters to tune. As well, the implementations of the tone mapping operators offer a tunable gamma value which is used as a post processing step. Including the weights mentioned above, the gamma values and the seven tone mapping algorithm parameters there are a total of 13 values to tune. This gives us a search space of 13 dimensions.

Each parameter has associated with it constraints which are used to keep the value of the parameter inside of the range which for it was intended. Table 3.1 gives the default setting for the parameters as well as the minimum and maximum values. The default value for gamma of 2.0 is taken from the implementation of the tone mapping operators. According to Reinhard et al. [34] this value is in the range of a typical display. The minimum and maximum values of 1.50 and 2.75, respectively, were chosen to span a range of gamma values typical of most display devices. The weights are mutated such that they are always positive between zero and one and

Operator	Parameter	Default	Min	Max
Schlick	$\ln(M)$	-0.74	-6.0	0.0
Ashikhmin	$\ln(s)$	-1.00	-2.0	4.0
	$t$	0.50	0.0	1.0
Reinhard & Devlin	$d$	1.00	0.0	1.0
	$c$	0.00	0.0	1.0
	$f$	0.00	-8.0	8.0
	$m$	0.60	0.3	1.0

Table 3.1: The default, minimum, and maximum values which are used for the tone mapping operators' input parameters. Default values are used for the parent of the first generation and the minimum and maximum values are used as constraints.

their sum is equal to one, which is done by dividing by the sum of the weights.

### 3.1.2 Generating Offspring

The first parent is generated using equal weights for each of the tone mapping operators, and the default parameter settings listed above. These values are all stored in the parent vector  $\mathbf{y}$ . New offspring are then generated as vectors  $\mathbf{x}$ , mutated from the single parent  $\mathbf{y}$ . This is done component wise by

$$x_i = (x_i^{(max)} - x_i^{(min)})\sigma r_i + y_i \quad (3.1)$$

where  $\sigma$  denotes the mutation strength or step size applied to the offspring,  $x_i^{(min)}$  and  $x_i^{(max)}$  are the minimum and maximum values for the parameters and weights which make up the candidate vectors. Lastly the  $r_i$  are normally distributed random numbers with mean 0 and standard deviation 1 and the  $y_i$  are the components of the parent vector. An exception will occur when a parameter,  $x_i$  in  $\mathbf{x}$ , has a range where the minimum and maximum values have orders of magnitude which differ by two or more. These parameters will be mutated as

$$x_i = \exp\left(\left(\ln(x_i^{(max)}) - \ln(x_i^{(min)})\right)\sigma r_i + \ln(y_i)\right). \quad (3.2)$$

This exception is in place to allow for parameters with large ranges to experience mutations similar to those with smaller ranges.

Constraint handling is done using the repair method of clamping the values of the parameters which are outside of the constraints. The clamping maps values greater

than the maximum to the maximum value, and values less than the minimum to the minimum value. This method of constraint handling was chosen as it is fast compared to rejecting offspring until  $\lambda$  offspring are generated with parameters which all fall into the acceptable ranges.

Step size adaptation is commonly used in evolutionary computation to guide the mutation towards an optimal solution much faster. For our tool we have decided to use a very simple strategy for adapting the step size of our mutations. This is mainly due to the fact that we do not expect a session to last more than 10 or 20 generations. Our step size is controlled by a mutation strength  $\sigma$  which starts out with a value of 0.2 and is decreased by one fifth each time a generation of images does not make any improvements and the parent image is selected. The parameter  $\sigma$  controls how much the parameters mutate away from the parent. In practice this works fairly well. However, depending on the image and the direction of the candidate vector, the variability of the offspring can be difficult to control. We will later address this issue by introducing a stage in which the offspring are first pre-screened such that the generation prepared for the user is all within a target range of perceptual similarity. This allows us to control the offspring in perceptual space rather than in parameter space.

### 3.1.3 Image Blending

The tone mapping operators give us three images for each offspring. With these images we use the mutated weights to blend the three together. This is done in the perceptually uniform colour space, CIELAB described in Section 2.3.3. The reason for using this space is that its perceptual uniformity should allow a good interpolation between the three pixel values. Moreover, our initial experimentations of using the RGB colour space for this interpolation led to some undesirable effects. These effects were most likely quantization errors near the extreme ends of the colour channels' ranges. Using the CIELAB colour space the issue of these undesirable effects was resolved. For the blending of the images, we first convert the image from RGB to CIELAB then each pixel,  $\mathbf{p}$  of the final image is calculated as

$$\mathbf{p} = w_a \mathbf{p}_a + w_s \mathbf{p}_s + w_{rd} \mathbf{p}_{rd} \quad (3.3)$$

where  $\mathbf{p}_a$ ,  $\mathbf{p}_s$  and  $\mathbf{p}_{rd}$  are corresponding pixels from the images tone mapped using Ashikhmin's, Schlick's and Reinhard and Devlin's operators respectively. Once the final blended image is processed, it is then transformed from CIELAB to XYZ by inverting the equations in Section 2.3.3. Lastly the image is transformed from XYZ to RGB using the inverse of the matrix from equation (2.2). The image is now ready to be displayed.

### 3.2 Perceptually Based Step Length Adaptation

The basic algorithm described so far is only capable of manipulating the offspring of a generation in parameter space which is done via the mutation strength,  $\sigma$ . It may on the other hand be desirable to be able to control the offspring in a perceptual space. In other words, rather than just having control over the mutations in parameter space we may want to have control over how similar the resulting images are when compared to one another visually. It may be noticeable, especially for some images and directions in parameter space, that small steps do not equate to small steps in perceptual space. Therefore, it may be desirable to be able to control the mutation of the offspring in perceptual space ensuring all offspring are roughly equidistant from the parent. This section will describe a method using an image similarity metric to control the mutation strength of the offspring.

Image space step size adaptation is used to control offspring in perceptual space. Here perceptual space will be described in terms of perceptual similarity which will be measured with the universal image fidelity metric described in Section 2.4.3. In order to be able to adapt offspring in terms of their perceptual similarity we will describe how to control offspring in a manner which will allow us to manipulate the similarity measure. We will also define a scheme to adapt the range of target perceptual similarity values to control the diversity of the population.

In order to produce generations of images where all of the images are within some target region of perceptual similarity around the parent image we conduct an initial pre-screening to adapt the images which do not meet our specifications. For the image metric we need to decide on the values of weights,  $w_L$ ,  $w_\alpha$  and  $w_\beta$ . We have chosen  $w_L$ ,  $w_\alpha$  and  $w_\beta$  to take on values of 3.00, 1.10 and 0.95, respectively. These are

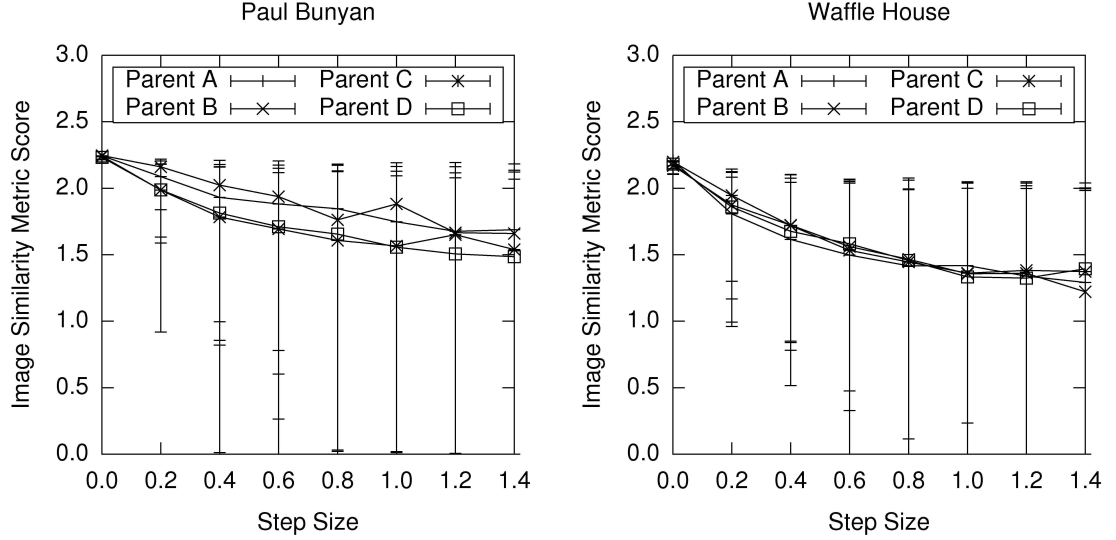


Figure 3.2: These plots show the average perceptual similarity of offspring from four different parents at different step sizes. The perceptual similarity has a range of  $[0.0, 2.24722]$  and the step size is varied from 0.0 to 1.4 with discrete steps of 0.2. The Paul Bunyan image can be seen in Figure 3.3 and the Waffle House image in Figure 3.11.

somewhat arbitrary but have been chosen to be close to the optimal ones reported by Toet and Lucassen [43] for their images. This gives us the equation

$$Q_{colour}(\mathbf{x}, \mathbf{y}) = \sqrt{3.00(Q(\mathbf{x}_L, \mathbf{y}_L))^2 + 1.10(Q(\mathbf{x}_\alpha, \mathbf{y}_\alpha))^2 + 0.95(Q(\mathbf{x}_\beta, \mathbf{y}_\beta))^2} \quad (3.4)$$

which is further described in Section 2.4.3. With these settings for the weights the metric has a range of  $[0.0, 2.24722]$ , where the maximum value of 2.24722 is achieved when the two images being compared are identical.

### 3.2.1 Perceptual Ranges

Prior to deciding whether it would be feasible to provide a feature which would allow the tool to generate images with a target perceptual similarity, some preliminary research was carried out. The goal here was to find a relationship between parameter space and perceptual space. By doing so this would allow us to predict the influence on the image in perceptual space when making changes to its parameter space. The results of the first experiment are plotted in Figure 3.2. This shows the average



Figure 3.3: This series of images shows how the similarity metric  $Q_{colour}$  behaves with respect to step size. The images here are the result of fixing a single random direction in parameter space and then generating an image at increasing step sizes. In this case we see that the further along the direction we travel away from the image the metric decreases.

perceptual similarity with 95% error bars of offspring at different step sizes. This was calculated for two different images and for each image four different random parents were chosen. As we can see, as the step size increases the average perceptual similarity decreases. As well we see the span of the error bars decrease when the step size is small. As well as showing the possible relationship between step size and the image similarity metric, this figure further emphasizes the need for greater control over the strength of the mutations. The large ranges covered by the error bars in the two plots show us that its difficult to predict the similarity of a new generated offspring solely based on the mutation strength. A second example shown in Figure 3.3 allows us to observe visual what is happening in Figure 3.2. This example shows a series of offspring all generated from a single parent using the same mutation vector. Each of these offspring differ by an increasing step size. As we can see in this example, as the step size increases the similarity metric decreases.

Figures 3.2 and 3.3 indicate that there is a fairly smooth relationship between step size and perceptual similarity. From this observation we make the assumption that by increasing the step size we can decrease the perceptual similarity and by decreasing the step size, increase the perceptual similarity. This assumption will allow us to control the perceptual similarity measure of the child images. We will use this result to help form a set of equations which will allow us to generate images within the desired perceptual range.

For the purpose of the pre-screening, the images will each have their own local mutation strength  $\sigma_l$ , initially set to the mutation strength  $\sigma$ , described in Section 3.1.1, at the start of each generation. The offspring will be generated as previously described, however using  $\sigma_l$  rather than the global mutation strength  $\sigma$ . Then the offspring will all be compared with the parent of the generation calculating its perceptual similarity,  $d$ . If an offspring is not within the range of perceptual similarities, denoted by  $r_{min}$  and  $r_{max}$ , then its local mutation strength will be decreased or increased, if the perceptual similarity of the offspring is below or above the range, respectively.

The local mutation strength is adapted by

$$\sigma_l = \begin{cases} \sigma_l(1.0 + |d - r_{max}|/2.24722) & : d > r_{max} \\ \sigma_l(1.0 - |d - r_{min}|/2.24722) & : d < r_{min} \\ \sigma_l & : \text{otherwise} \end{cases} \quad (3.5)$$

The idea is to decrease or increase the mutation strength relative to how far outside the similarity range the offspring is. This process is repeated until all of the offspring of a generation fall into the target range.

### 3.2.2 Perceptual Similarity Ranges

We would like to define a range of perceptual similarities which will control the offspring that will be selected as candidates for the generation. Figure 3.4 offers insights regarding which ranges of perceptual similarity will allow us to produce a generation of images in a feasible amount of time. The idea here is to start with a rather large range with a significant distance from the parent. Then as the range is adapted it should converge quickly to a range not too small but such that it will generate images very close to the parent.

The range has a maximum of  $\sqrt{5.05}$  which is approximately equal to 2.24722 and minimum of 0. The application stores a value for the target perceptual similarity,  $t$ , and with this we are able to determine with a range of perceptual similarities,  $[r_{min}, r_{max}]$ . This range is calculated by

$$r_{min} = t - (1 - (t/t_{max})^{1.5} + m_{min}) \quad (3.6)$$

and

$$r_{max} = t + (1 - (t/t_{max})^{1.5} + m_{min}) \quad (3.7)$$

where  $m_{min}$  has been set to 0.1, which keeps the interval open so that it does not converge to a single value. The factor  $1 - (t/t_{max})^{1.5}$  allows the range to start off fairly large and then quickly decrease as it approaches the final range. The updates will occur whenever the parent of the current generation is selected as the best available candidate, the same as in the parameter space step size adaptation. These updates are calculated as

$$t = t + (t_{max} - t)\epsilon \quad (3.8)$$



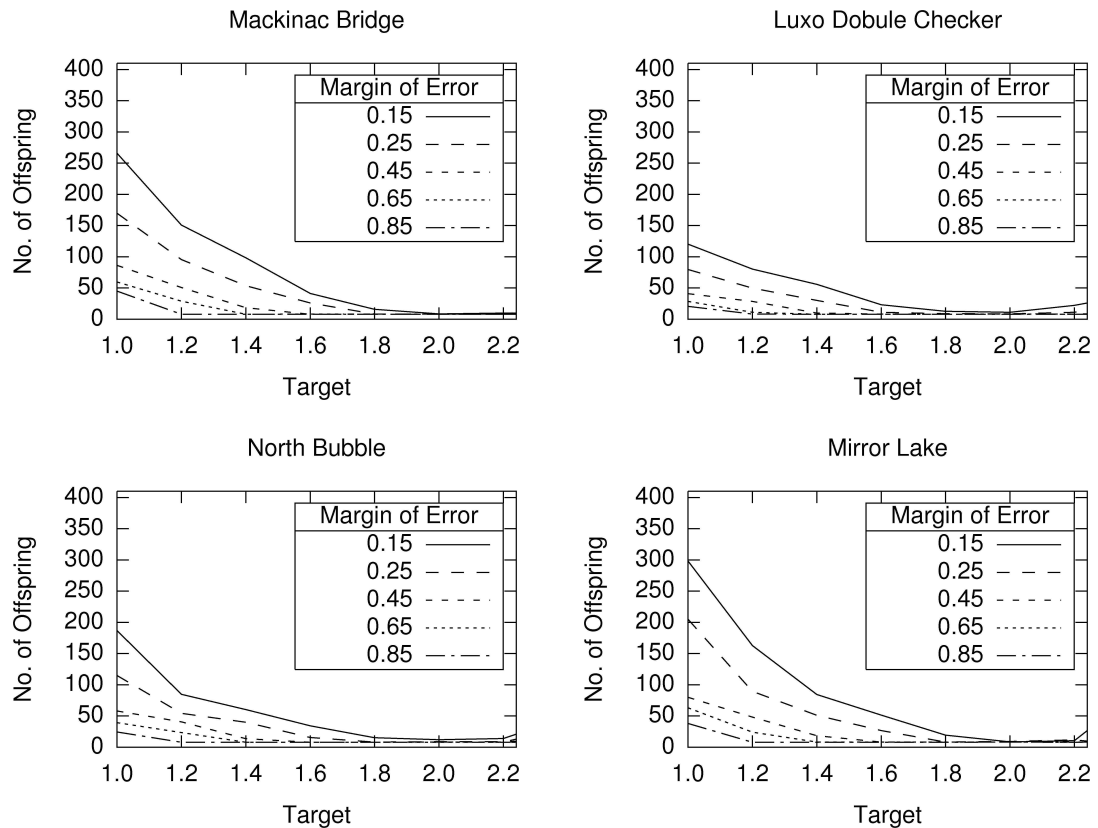


Figure 3.4: This figure shows the average number of images generated for different perceptual similarity targets and margins of error in order to generate eight child images within the range defined by the target and margin. The offspring for this plot were all generated using an initial step size of 0.2. This was done for four different images from [11]. These results were created to support the decisions regarding the adaptation of the perceptual ranges.

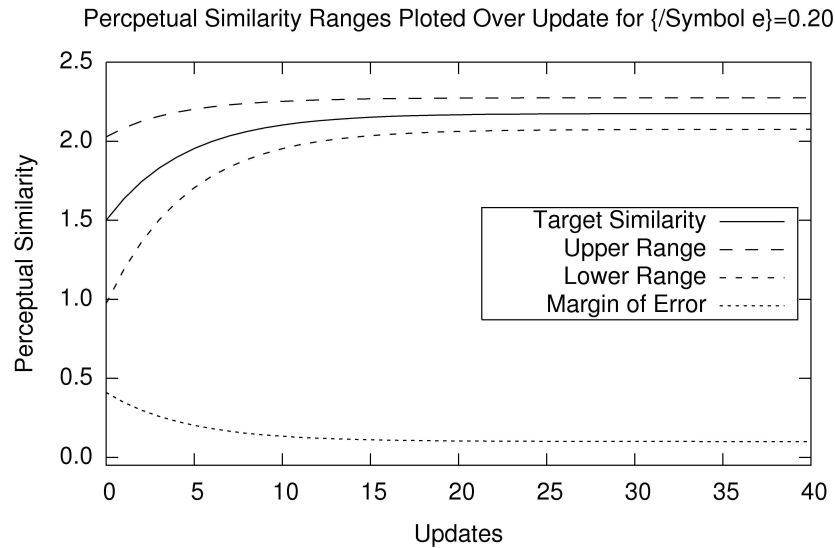


Figure 3.5: This shows how the range of perceptual similarities will adapt with each update which occurs. The maximum line shows the maximum value which the similarity metric can obtain. The margin of error and target lines show how each of these are adapted to define the range of similarities.

where  $t_{max}$  has been set to 2.175. This allows the range to start off approximately spanning  $[1.0, 2.0]$  and then will approach a range of approximately  $[2.1, 2.25]$ . By adjusting the value for  $\epsilon$  we are able to control how fast the range approaches final values. Figure 3.5 shows a plot of the range adaptation with  $\epsilon = 0.20$ . As we can see the range is very close to the final values after just 10 updates which is likely a good setting for our application since we do not expect the user to endure numerous generations.

Since the diversity of the offspring is controlled by a band of perceptual similarities it is reasonable to increase the initial step size which images will be generated with. This way if offspring which are drastically different from the parent, they should then be rejected by the perceptual similarity range and adapted to fit.

### 3.2.3 Speed Enhancements

One of the constraints that we are working with is speed. As described in Section 2.6, user fatigue is one of the most common issues with interactive evolution. Given this we would like to maximise the number of generations a user is able to endure

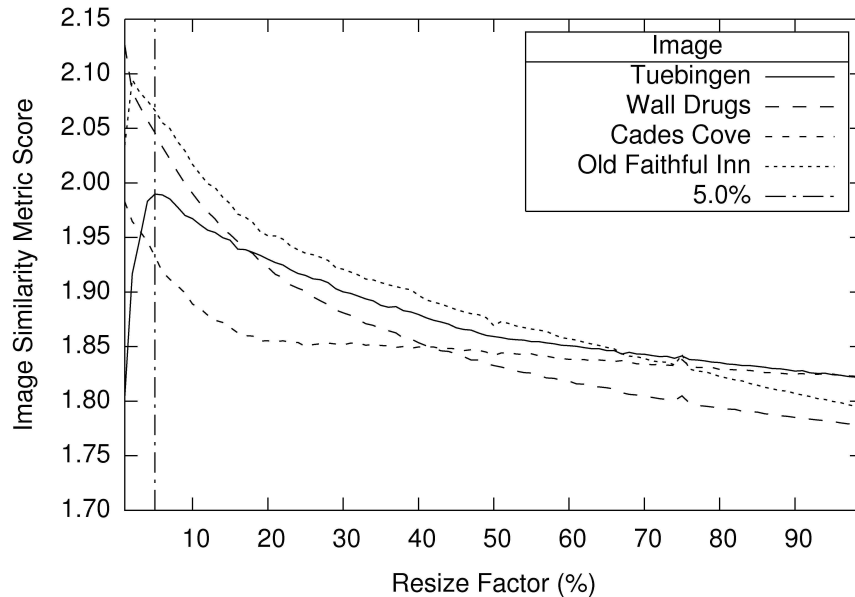


Figure 3.6: This figure shows the relationship between the size of the image and perceptual similarity measured with the universal colour image fidelity metric. Four pairs of images with an initial size of  $1024 \times 683$  are compared after being resized from 99% to 1%. The resizing was done using the box filter as it left the similarity metric the least affected. From the plot we can see that until the 5% line all of the curves increase gradually. At this point the images have a resolution of  $50 \times 34$ .

and so decreasing the time between generations is the next topic of discussion. There are several areas where speed can be increased using common tools. Multi-threading is one technique which we use to decrease the time it takes to produce the next generation. This takes advantage of multiple cores found in modern processors. All of the tone mappings which take place in each generation can be run in parallel, as well as the blending. This is very useful as at least eight offspring are generated in each generation and each of these needs three tone mapping operations and one blending operation. This totals at least 24 tone mappings and eight blending operations. This total can be even higher when the goal is to generate offspring within a range perceptual similarities as offspring are repeatedly generated until there are enough acceptable offspring.

Secondly, in order to produce a generation of images which all lie in the same perceptual similarity range, images are pre-screened. New batches of offspring are generated until there are eight offspring which fall into the target range of perceptual

similarities. The initial pre-screening is done with much smaller images. Although smaller images take less time to process, the amount of information has changed and so this can affect the score of the image metric. Tests were performed to find the optimal scaling factor which would allow us to find images in the correct range without requiring too much time between generations. Figure 3.6 shows the relationship between image size in percent and perceptual similarity scores. This is shown for a images with an original size of  $1024 \times 683$ . Other filters were tested but it was found that the box filter did the least to affect the similarity metric. We can see that by restricting the image resizing such that the resized image is no smaller than  $50 \times 34$  pixels we are able to obtain a good relationship between the score of the original sized images and the resized images.

### 3.3 Results

We now analyze and discuss the interactive evolutionary tone mapping tool using a few example sessions. First we look at a couple of sessions to make observations regarding how the two types of step size adaptation are able to provide diversity when the steps are large and then their abilities to tighten control over diversity as the steps become much smaller. This will provide insight into the possible usefulness of these methods and this tool as a whole.

Next will look at a sample session first without perceptual similarity ranges, comparing of the default settings for the three tone mapping operators and their blended image against the final settings and their blended image. We will then observe a sample session which uses the perceptual similarity ranges, looking again at the default and final settings. Also we will examine how the weights are mutated during these sessions. We will conclude this section with a comparison of a result mapped with the interactive evolutionary tone mapping tool and other published mappings of the same image.

#### 3.3.1 Offspring Step Size Control

For these two sessions we tone map the Golden Gate Bridge image [11]. In this image it is important to capture the detail from the shaded grassy area near the bottom of

the image as well as the detail from the rest of the image which, for the most part, is much brighter. These two sessions allow us to observe how the step size adaptation controls the images. In each session the step size is has been reduced five times, the parent image is displayed in the middle of its eight children surrounding it. The goal is to observe a wide diversity before the step size is reduced and then a much less diverse group of offspring after the step size has been reduced. The number of step size reductions was chosen as five, since typically during a session lasting 10 to 20 generations the step size should be decreased about five times.

First we will examine the starting generation, shown in Figures 3.7 and 3.8. These were produced with similar parent images, which are placed in the centre of each of these figures. The offspring in the centre of the bottom row of Figure 3.7, which uses parameter space adaptation, is obviously not a good mapping. The point of using the ranges of perceptual similarity is to attempt to remove such images from generations as well as to maintain control over diversity with small step sizes. As we can see from Figure 3.8 there are no images which are mapped as poorly as the one in the centre on the bottom row in Figure 3.7, yet we are still able to see some variety in the offspring.

After adapting the step size five times, the parameter space step size adaptation which started with a step size of 0.2, ended with a step size of 0.066. The perceptual similarity step size adaptation started with a step size of 0.3, and ended with the step size 0.0988. As well the perceptual similarity step size adaptation ended with a perceptual similarity range between 1.705 and 2.203.

For Figures 3.9 and 3.10 the parent images in the centre have remained the same. The difference here is that the step sizes have been reduced according to the functions discussed above. The idea now is to have very little diversity, yet a generation of completely identical offspring is not very useful either. Except for the image in the centre on the right column in Figure 3.9, the offspring have only subtle variations from the parent. The perceptual similarity ranges appear to have eliminated any drastic changes from the parent as the images in Figure 3.10 have only subtle variations.



Figure 3.7: This is the initial generation of images for the sample session using parameter step size adaptation. The image seen here and in the follow three figures is an image of the Golden Gate Bridge found in [11].



Figure 3.8: This is the initial generation of images for the sample session using perceptual space step size adaptation. We can see that there are no images similar to the one found in the centre of the bottom row of Figure 3.7.

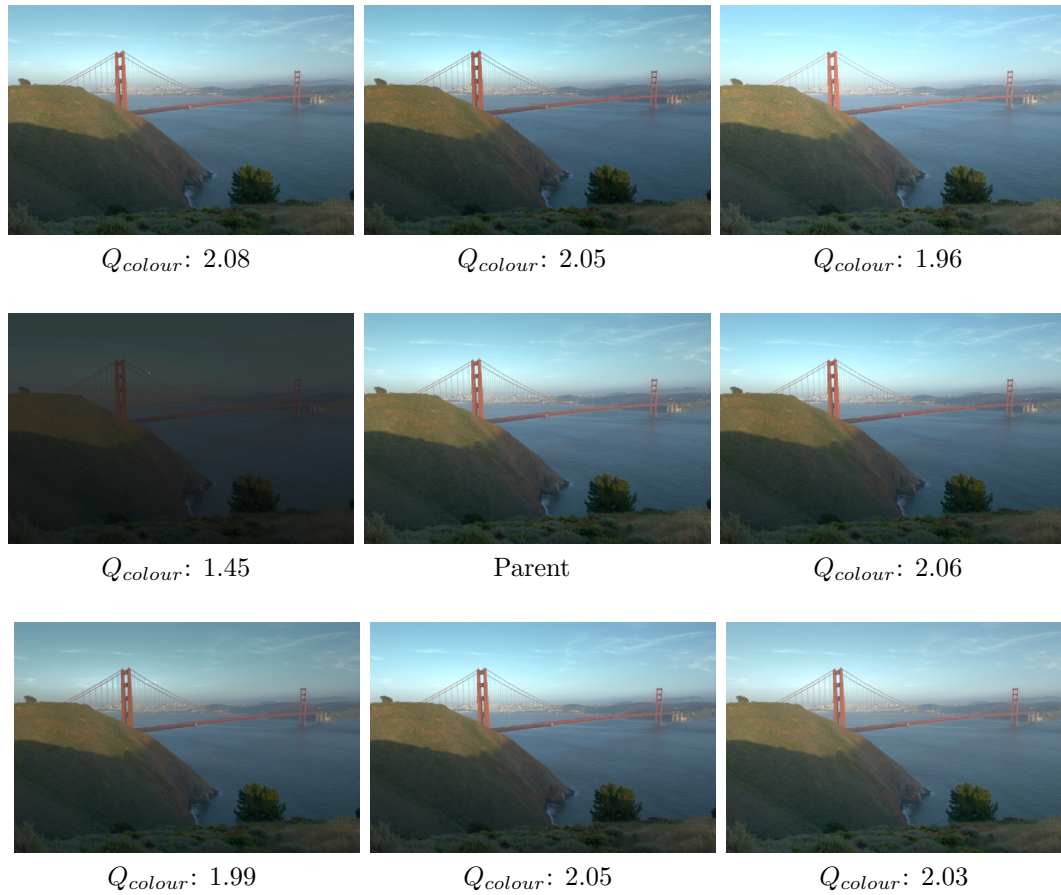


Figure 3.9: This is the final generation of images for the sample session using parameter space step size adaptation. We can see that even though the step size has been reduced there is still at least one image which is still fairly different from the parent image in the centre.



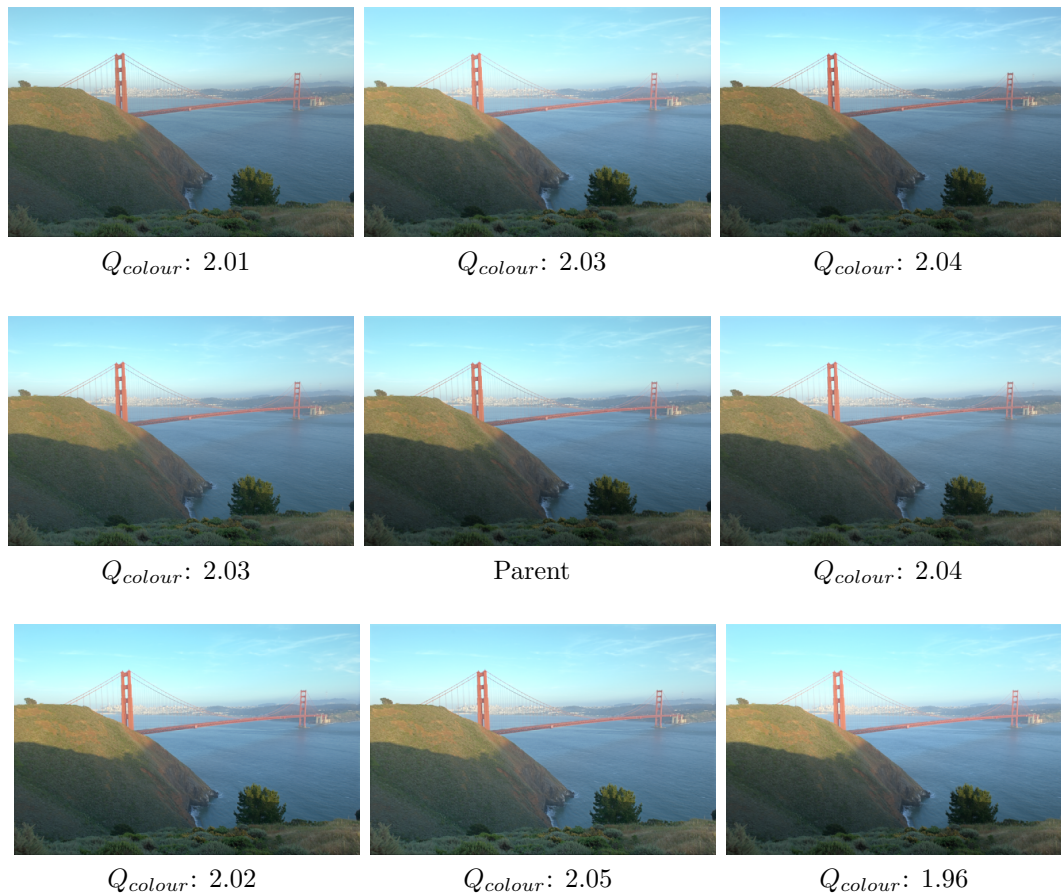


Figure 3.10: This is the final generation of image for the sample session using perceptual space adaptation. We can see here that by restricting the images in perceptual space we are able to produce a generation of very similar offspring.

### 3.3.2 Parameter Space Step Size Adaptation

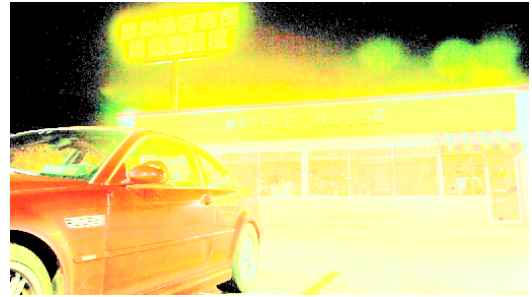
For this session and the next we will be looking at the Waffle House image [11]. The attractive feature of the Waffle House image for these observations is that neither of the tone mappings provides a perfect solution to start with. Looking at the default settings in Figure 3.11, Ashikhmin's operator does the best to preserve contrast. However, if one looks closely at the lights inside of the diner, halos are observed. Schlick's operator produces a very bright result which is quite unnatural and much of the detail in the image is missing. Lastly, Reinhard and Devlin's operator produces a poor result due to some bright spots which emerge in the high contrast areas around the sign outside the building. The blended result is not good due to these issues found in each of the images.

Selections were made during the interactive evolutionary process with the goal of reducing or removing the bright spots which appear around the top of the building. As well the overall image should not be too dark such that the detail in the car is missing. Also the halos around the lights inside the diner are additional artifacts which should be avoided.

After 12 generations we arrive at an arguably much better image than the default blended image as we can see in Figure 3.11. The lower weights on the results from Schlick's and Reinhard and Devlin's operators resulted in their contributions not mutating towards a something much better than what they started with. However, Ashikhmin's contribution which started out with the best result mutated toward a darker image which contains a lot of contrast. The weights which are plotted in Figure 3.12 show how each operator contributed to the final image. Ashikhmin's operator gives us the greatest contribution adding a lot of contrast, Reinhard and Devlin's operator contributes some brightness and Schlick's with a weight of zero or very close to zero contributed nothing. The plot in Figure 3.12 also shows us that the weights mutated quickly such that Ashikhmin's contribution dominated. This would result in most of the tuning of the parameters tuned this one operator. This is not to say that Schlick's, or Reinhard and Devlin's operator are completely inappropriate for this image. Better mappings could be obtained by manually tuning the parameters for each of the operators.



(a) Default;  
Schlick



(b) Final;  
Schlick



(c) Default;  
Ashikhmin



(d) Final;  
Ashikhmin



(e) Default;  
Reinhard &  
Devlin



(f) Final;  
Reinhard &  
Devlin



(g) Default;  
Blended



(h) Final;  
Blended

Figure 3.11: A sample session with parameter space step size adaptation using the Waffle House image. The final blended image does a good job at eliminating the bright spots around the sign while giving a good sense of the brightness of the sign.

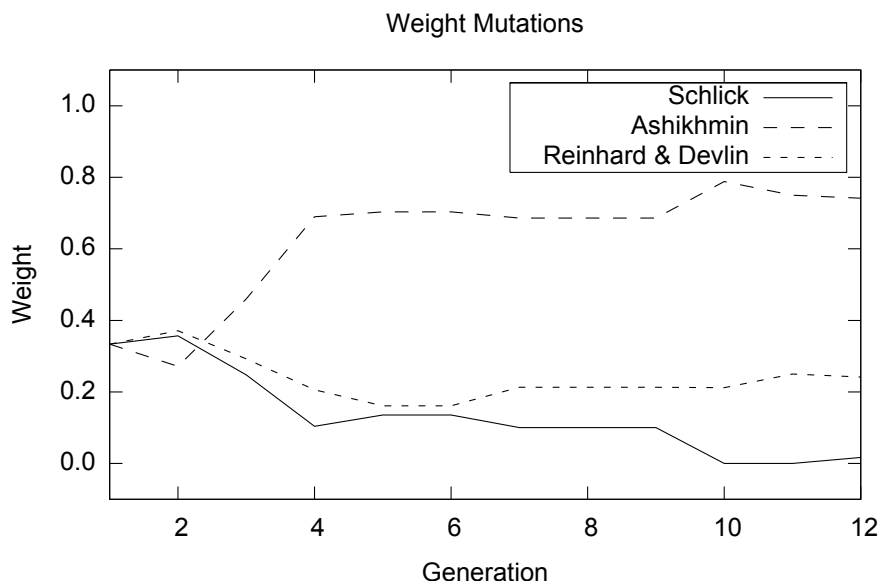


Figure 3.12: The plot shows how the weights adapt over the generations with parameter space step size adaptation. The places in the plot where all three of the weights do not change between generations indicate areas where the parent was selected.

### 3.3.3 Perceptual Space Step Size Adaptation

This session we will look at how perceptual similarity ranges work to adapt the starting image towards a final solution. This session will use the same default settings and thus images as the previous. The goals in the tone mapping remain the same.

The results of this session can be seen in Figure 3.13. This session resulted in a very similar mapping as that seen in Figure 3.11. As in the previous session the weight for Ashikhmin’s operator dominated over the rest quite quickly as shown in the plot in Figure 3.14. Again this is likely due to the fact that Ashikhmin’s operator has the best default mapping when compared to others. The final result here has both Reinhard and Devlin’s and Schlick’s operators contributing a small but noticeable amount. Schlick’s very bright mapping adds brightness to the final blended image and Reinhard and Devlin’s mapping helps to smooth some of the areas where halos are present in Ashikhmin’s mapping.

We will conclude the results section by comparing the results of an image tone mapped with our interactive evolutionary tool with some other published results. In Figure 3.15 we compare our results of mapping the Big Fog image tone mapped with



(a) Default;  
Schlick



(b) Final;  
Schlick



(c) Default;  
Ashikhmin



(d) Final;  
Ashikhmin



(e) Default;  
Reinhard &  
Devlin



(f) Final;  
Reinhard &  
Devlin



(g) Default;  
Blended



(h) Final;  
Blended

Figure 3.13: A sample session with perceptual space step size using the Waffle House image. The final blended image is quite similar to the previous sample session however the final blended image is somewhat brighter overall, this is likely due to the larger weighting given to Schlick's operator.

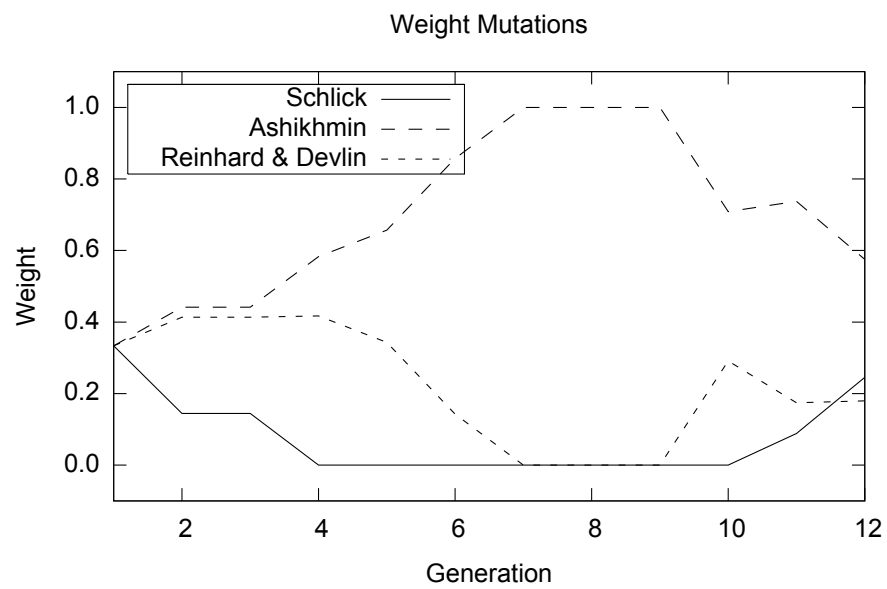


Figure 3.14: This plot shows the mutations of the weights for a sample session similar to the previous except this time with perceptual space step size adaptation. The weights here are very much the same as the previous session except for the weight associated with Schlick's operator which is larger in this session. This could be what is attributed to the brighter final blended image.



Figure 3.15: Comparison with other published results. The mapping on the left is a result from Fattal et al. [14] and the mapping on the right is from Ward Larson et al. [22]. The centre mapping was generated with the interactive evolutionary tone mapping tool. The high dynamic image is due to Tumblin and Turk [44].

the against results published by Ward Larson et al. [22] and Fattal et al. [14]. While it is not immediately clear which of the tone mapped images is preferable, the evolutionary approach allows each user to select according to their own preferences without manually changing obscure parameters. Figures 3.16 and 3.17 give more comparisons with other published results. In each of these we further see the subjective nature of tone mappings, strengthening the argument for using interactive evolution as a computational tool.

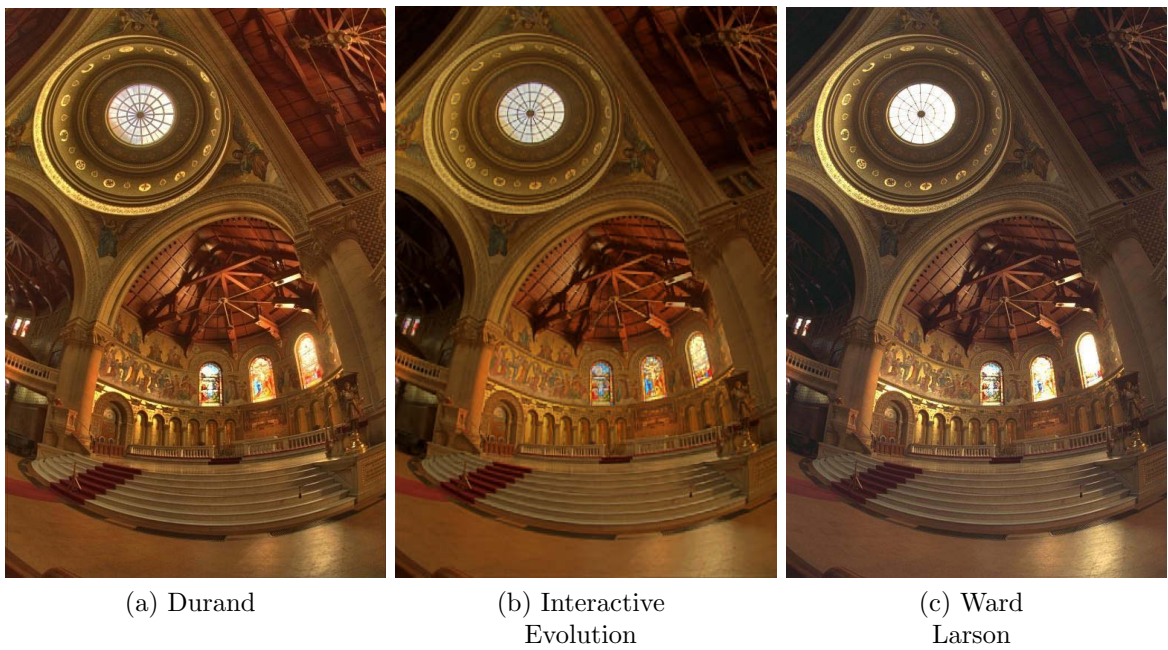


Figure 3.16: A second comparison with other published results. The mapping on the left is a result from Durand and Dorsey’s bilateral filtering algorithm [10] and the mapping on the right is from Ward Larson et al. [22]. The centre mapping was generated with the interactive evolutionary tone mapping tool. The high dynamic range image is included with [34].

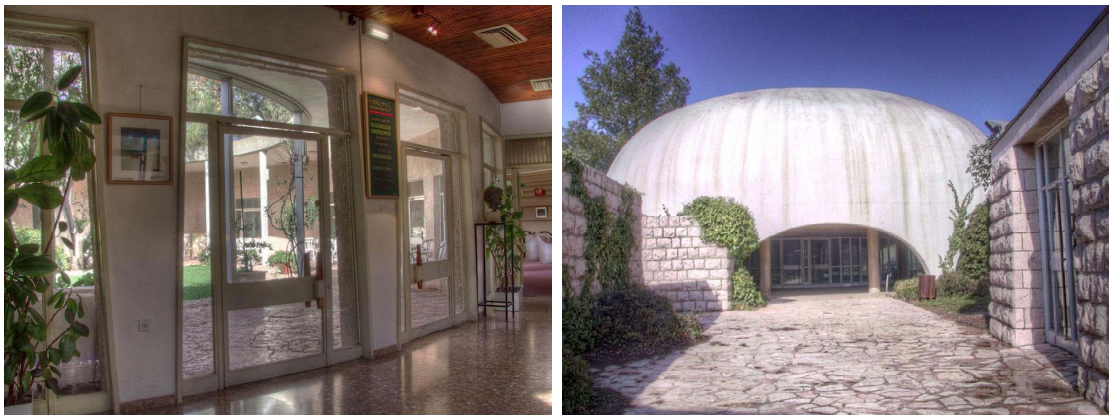




(a) Trilateral Filtering



(b) Interactive Evolution



(c) Fattal

Figure 3.17: Third and final comparison with other published results. The mappings on the top are two result from Choudhury and Tumblin's trilateral filtering algorithm [6]. On the bottom are results from Fattal et al. [14] and the mappings in the centre are results from the interactive tone mapping tool. Both of these images are due to Fattal et al. [14].

## Chapter 4

### Conclusion

In this thesis a new approach has been described that uses interactive evolution as a computational tool for tone mapping high dynamic range images. It is difficult to mathematically describe a good mapping yet for a human observer the task is almost effortless. Blending the results from several tone mapping operators allows for the tool to render mappings outside the capabilities of just one operator. Combining the parameters of each tone mapping operator and the blending weights into a single vector creates an optimization problem which uses subjective selection. The use of a perceptually uniform colour space to perform the blending proved necessary to avoid the difficulties of interpolating colour in RGB. Even though we are using a fairly simple  $(1 + \lambda)$ -ES, the system has been found to be capable of generating good mapping in just a few generations. The use of a perceptual similarity metric to control the offspring of a generation presents a novel idea for interactive evolution on images. The application created for this thesis will be released to the public as a tool for tone mapping.

In future work it would be desirable to conduct an in depth set of user experiments to examine the strengths and weaknesses of our approach. This would include a study where participants would be asked to tone map a high dynamic range image using the tool with a goal of producing a mapping similar to a target. This would provide further insights into the usefulness of the approach as well as help to find optimal values for step length adaptation for both  $\sigma$  and the perceptual ranges. Giving control over the step length to the user may also prove useful. In addition, valuable data could be collected from multiple runs from many users of the tool, this could be collected through an online setting of the tool. This may allow us to determine if a covariance matrix adaptation method would be useful with this tool. If so this could decrease the number of generations needed in order to find an appropriate mapping.

## Bibliography

- [1] M. Ashikhmin. A tone mapping algorithm for high contrast images. In *EGRW '02: Proceedings of the 13th Eurographics Workshop on Rendering*, pages 145–156. Eurographics Association, 2002.
- [2] W. Banzhaf. Interactive evolution. In T. Bäck et al., editors, *Handbook of Evolutionary Computation*. Oxford University Press, 1997.
- [3] H.-G. Beyer and H.-P. Schwefel. Evolution strategies — A comprehensive introduction. *Natural Computing*, 1(1):3–52, 2002.
- [4] R. Breukelaar, M. Emmerich, and T. Bäck. On interactive evolution strategies. In F. Rothlauf et al., editors, *EvoWorkshops 2006*, pages 530–541. Springer Verlag, 2006.
- [5] S. B. Chisholm, D. V. Arnold, and S. Brooks. Tone mapping by interactive evolution. In *GECCO 2009: Proceedings of the 11th Annual Conference on Genetic and Evolutionary Computation*, pages 515–522. ACM Press, 2009.
- [6] P. Choudhury and J. Tumblin. The trilateral filter for high contrast images and meshes. In *Proceedings of the Eurographics Symposium on Rendering*, pages 186–196, 2003.
- [7] P. Debevec, E. Reinhard, G. Ward, and S. Pattanaik. High dynamic range imaging. In *SIGGRAPH '04: ACM SIGGRAPH 2004 Course Notes*, page 14. ACM, 2004.
- [8] 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. ACM Press, 1997.
- [9] F. Drago, W. L. Martens, K. Myszkowski, and H.-P. Seidel. Perceptual evaluation of tone mapping operators. In *SIGGRAPH '03: ACM SIGGRAPH 2003 Sketches & Applications*. ACM Press, 2003.
- [10] 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. ACM Press, 2002.
- [11] M. D. Fairchild. *The HDR Photographic Survey*. MDF, 2008.

- [12] M. D. Fairchild and G. M. Johnson. Meet iCAM: A next-generation color appearance model. In *IS&T/SID 10th Color Imaging Conference*, pages 33–38, 2002.
- [13] M. D. Fairchild and G. M. Johnson. The iCAM framework for image appearance, image differences, and image auality. *Journal of Electronic Imaging*, 13(1):126–138, 2004.
- [14] R. Fattal, D. Lischinski, and M. Werman. Gradient domain high dynamic range compression. *ACM Transactions on Graphics*, 21(3):249–256, 2002.
- [15] J. Foley, A. van Dam, S. Feiner, and J. Hughes. *Computer Graphics: Principles and Practice (Second Edition in C)*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1997.
- [16] A. S. Glassner. *Principles of Digital Image Synthesis*. Morgan Kaufmann Publishers Inc., 1994.
- [17] J. Graf and W. Banzhaf. Interactive evolution of images. In *Evolutionary Programming IV: Proceedings of the Fourth Annual Conference on Evolutionary Programming*, pages 53–65. MIT Press, 1995.
- [18] M. Herdy. Evolution strategies with subjective selection. In *PPSN IV: Proceedings of the 4th International Conference on Parallel Problem Solving from Nature*, pages 22–31. Springer Verlag, 1996.
- [19] M. Herdy. Evolutionary optimisation based on subjective selection — evolving blends of coffee. In *EUFIT '97: Proceedings of the 5th European Congress on Intelligent Techniques and Soft Computing*, pages 640–644, 1997.
- [20] G. Johnson and M. Fairchild. Rendering hdr images. In *IS&T/SID 11th Color Imaging Conference*, pages 36–41, 2003.
- [21] J. Kuang, H. Yamaguchi, C. Liu, G. M. Johnson, and M. D. Fairchild. Evaluating HDR rendering algorithms. *ACM Transactions on Applied Perception*, 4(2), 2007.
- [22] G. Ward Larson, H. Rushmeier, and C. Piatko. A visibility matching tone reproduction operator for high dynamic range scenes. *IEEE Transactions on Visualization and Computer Graphics*, 3:291–306, 1997.
- [23] P. Ledda, A. Chalmers, T. Troscianko, and H. Seetzen. Evaluation of tone mapping operators using a high dynamic range display. *ACM Transactions on Graphics*, 24(3):640–648, 2005.

- [24] J. Marks, B. Andalman, P. A. Beardsley, W. Freeman, S. Gibson, J. Hodgins, T. Kang, B. Mirtich, H. Pfister, W. Ruml, K. Ryall, J. Seims, and S. Shieber. Design galleries: a general approach to setting parameters for computer graphics and animation. In *SIGGRAPH '97: Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques*, pages 389–400. ACM Press, 1997.
- [25] F. Mccollough. *Complete Guide to High Dynamic Range Digital Photography*. Lark Books, Asheville, 2008.
- [26] Z. Michalewicz. A survey of constraint handling techniques in evolutionary computation methods. In *Proceedings of the 4th Annual Conference on Evolutionary Programming*, pages 135–155. MIT Press, 1995.
- [27] Z. Michalewicz and N. Attia. Evolutionary optimization of constrained problems. In *Proceedings of the 3rd Annual Conference on Evolutionary Programming*, pages 98–108. World Scientific Publishing, 1994.
- [28] C. Neufeld, B. J. Ross, and W. Ralph. The evolution of artistic filters. In Juan Romero and Penousal Machado, editors, *The Art of Artificial Evolution: A Handbook on Evolutionary Art and Music*, pages 335–356. Springer Berlin Heidelberg, 2007.
- [29] M. Pharr and G. Humphreys. *Physically Based Rendering*. Morgan Kaufmann, 2004.
- [30] D. Powell and M. M. Skolnick. Using genetic algorithms in engineering design optimization with non-linear constraints. In *Proceedings of the 5th International Conference on Genetic Algorithms*, pages 424–431. Morgan Kaufmann Publishers Inc., 1993.
- [31] E. Reinhard, M. Ashikhmin, B. Gooch, and P. Shirley. Color transfer between images. *IEEE Computer Graphics and Applications*, 21(5):34–41, 2001.
- [32] E. Reinhard and K. Devlin. Dynamic range reduction inspired by photoreceptor physiology. *IEEE Transactions on Visualization and Computer Graphics*, 11(1), 2005.
- [33] E. Reinhard, M. Stark, M. Shirley, and J. Ferweda. Photographic tone reproduction for digital images. In *SIGGRAPH '02: Proceedings of the 29th Annual Conference on Computer Graphics and Interactive Techniques*, pages 267–276. ACM Press, 2002.
- [34] E. Reinhard, G. Ward, S. Pattanaik, and P. Debevec. *High Dynamic Range Imaging: Acquisition, Display, and Image-Based Lighting*. Morgan Kaufmann, 2005.

- [35] D. L. Ruderman, T. W. Cronin, and C.-C. Chiao. Statistics of cone responses to natural images: Implications for visual coding. *Journal of the Optical Society of America A*, 15:2036–2045, 1998.
- [36] C. Schlick. Quantization techniques for visualization of high dynamic range pictures. In G. Sakas et al., editors, *Photorealistic Rendering Techniques*, pages 7–20. Springer Verlag, 1995.
- [37] J. Secretan, N. Beato, D. B. D’Ambrosio, A. Rodriguez, A. Campbell, and K. O. Stanley. Picbreeder: Collaborative interactive evolution of images. *Leonardo*, 41(1):98–99, 2008.
- [38] J. Secretan, N. Beato, D. B. D’Ambrosio, A. Rodriguez, A. Campbell, and K. O. Stanley. Picbreeder: Evolving pictures collaboratively online. In *CHI ’08: Proceedings of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, pages 1759–1768. ACM Press, 2008.
- [39] G. Sharma, M. J. Vrhel, and H. J. Trussell. Color imaging for multimedia. In *Proc. IEEE*, pages 1088–1108, 1998.
- [40] K. Sims. Artificial evolution for computer graphics. *Computer Graphics*, 25(4):319–328, 1991.
- [41] W. S. Stiles and J. M. Burch. Interim Report to the Commission Internationale de l’Eclairage, Zurich, 1955, on the National Physical Laboratory’s Investigation of Colour-matching (1955). *Optica Acta: International Journal of Optics*, 2(4), 1955.
- [42] H. Takagi. Interactive evolutionary computation: fusion of the capabilities of EC optimization and human evaluation. *Proceedings of the IEEE*, 89(9):1275–1296, 2001.
- [43] A. Toet and M. P. Luscassen. A new universal colour image fidelity metric. *Displays*, 24(4-5):197–207, 2003.
- [44] 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. ACM Press/Addison-Wesley Publishing Co., 1999.
- [45] Z. Wang and A. C. Bovik. A universal image quality index. *IEEE Signal Processing Letters*, 9(3):81–84, 2002.
- [46] Z. Wang and A. C. Bovik. Mean squared error: Love it or leave it? - a new look at signal fidelity measure. *IEEE Signal Processing Magazine*, 26(1):98–117, 2009.

- [47] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.
- [48] 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. ACM, 2004.
- [49] A. Wiens and B. J. Ross. Gentropy: Evolutionary 2d texture generation. *Computers and Graphics*, 26(1):75–88, February 2002.