Printing HDR Images:
A Strategy Using Multiple Transparencies

Submitted in partial fulfilment
of the requirements of the degree
Bachelor of Science (Honours)
of Rhodes University

Benji Euvrard

November 2009

Abstract

In this research paper we propose methods of printing High Dynamic Range (HDR) images on print media. We do this by printing on multiple transparencies, using a bright backlight to shine through the transparencies. In this way we overcome the limitations of relying on reflectance to reproduce a desired luminance. Designing and using a light meter, we first calibrate the printer for one transparency before applying an additive, and separately a multiplicative function to model the attenuation of the backlight based on the combination of colours on the transparencies. We then try models based on the same shade on each transparency, per pixel, and different shades, combining to represent the desired shade and luminosity. After evaluating our different models, we conclude that the model based on a multiplicative function modelling the attenuation of the backlight, with different shades on each transparency most accurately portrays the original HDR image, with the dynamic range reproduced proportional to the number of transparencies.

Acknowledgements

A special thanks to my parents, my supervisor Shaun (Professor Shaun Bangay) and the VRSIG group: Matt, Matt, Ross and Colin. I must acknowledge the financial and technical support of this project of Telkom SA, Comverse SA, Stortech, Tellabs, Amatole, Mars Technologies, Bright Ideas Projects 39 and THRIP through the Telkom Centre of Excellence at Rhodes University.

Contents

Chapter 1

Introduction

1.1 Problem Statement

We propose several methods of printing High Dynamic Range (HDR) images, while still retaining the dynamic range. This involves solving a number of problems:

1. Creating HDR images
2. Relating the light intensities captured by the camera to the quantities of ink produced by a printer by:
   (a) Calibrating the response of printers across their entire range.
   (b) Building and using a light meter to measure printer response.
3. Measuring the dynamic range of a printed image.

By devising methods of converting an HDR image to multiple transparencies, using a bright backlight, we can achieve our goal of retaining a higher dynamic range of the original image, than when printing using conventional methods. The main principle of our devised method is to separate an HDR image onto multiple transparencies, such that when combined, the ink on the transparencies will attenuate the light such that both colour and luminosity are more accurately reproduced. To understand the strategies proposed, and why they are proposed, we provide a short background to HDR imaging along with a review of related work in Chapter 2. After this we go about calibrating the printer for one transparency, building and using a light meter before describing a few methods. We then evaluate these methods and draw some conclusions from our evaluation.

1.2 Background

In a conventional 8-bit colour image file, each pixel contains 8 bits of information per colour channel. Each pixel, therefore, can hold 24 bits of information (i.e. can be 1 of 2^{24} =16 777 216 colours). An HDR image, on the other hand, stored in the OpenEXR format can hold 32 bits of information per colour channel, so each pixel can be 1 of 2^{96} =79 228 162 514 264 337 593 543 950 336 colours. An HDR image can, therefore, hold a greater dynamic range: the ratio between the lightest and darkest pixel (Ledda et al. [12], McCollough [14], Trentacoste [23]). This does not, however, mean that an OpenEXR file contains a greater dynamic range than an 8 bit image file, merely that it has the capacity to do so i.e. the range of intensities is still about 8 bits worth.

To create an HDR image, several photos of the same scene are taken, at different exposures (McCollough [14]). This is to ensure that all details of the scene are recorded: the detail of the darker areas recorded in the over-exposed photos; the details of the lighter areas recorded in the under-exposed photos. These photos are then merged into a single HDR image, using this wide range of detail, using software such as Adobe Photoshop or Photomatix.

Chapter 2

Background and Related Work

2.1 Introduction

In this section, a background to HDR images is discussed. Related work is presented, describing current techniques for displaying them in both on print media and electronic displays. This includes a brief introduction as to what exactly HDR images are (Section 2.1.1), how they are created (Section 2.1.3), and problems involved in working with them (Section 2.1.4). Various techniques are discussed on tackling these problems (Sections 2.2 and 2.3). And finally printing techniques (Section 2.4) are discussed. This section lays a foundation for developing a process to print HDR images.

2.1.1 What are HDR images?

HDR images are images with a broader range of luminances (i.e. amount of light) between light and dark than regular images (Ledda et al. [12], McCollough [14], Trentacoste [23]). There is research into redefining what exactly dynamic range means to more closely match human perception, tackling issues such as ambient lighting and usefulness of minimum darkness values (Ward [25]). Ward [25] considers, a new method for defining dynamic range, using the Number of Distinguishable Grays (NDG). He answers the question, "How many gray levels will a standard observer be able to distinguish between the darkest black and the brightest white in a given ambient environment?" The discussion of defining dynamic range, however, is not discussed in this section.

An HDR image can be created and manipulated using appropriate software (e.g. Adobe Photoshop, FDRTools, GIMP, Photomatix). Creating HDR images is discussed in Section 2.1.3. The HDR image is then stored in a 32-bit format using one of a variety of file types: HDR, OpenEXR, PSD, or floating point TIFF.

2.1.2 Why have HDR images?

Real world scenes contain a huge range of light intensities which the human visual system (HVS) is efficient at adapting to see (Howard [8], Ward et al. [28], Nayar and Mitsunaga [15]). What we see in a photograph is very different to what we perceive at the time of that the photograph is taken (Agarwala et al. [2], Ledda et al. [12], Wells [29]). One reason is that our brain fills in details, seeing more than just the moment in time that a camera sees (Agarwala et al. [2]). A more relevant reason is that cameras are unable to capture enough dynamic range. That is, they fail to capture enough detail in both the bright and dark areas of a scene. Therefore, people such as Mann [13] and DebevecFerwerda and Malik [4] conduct research into combining multiple photographs to create an image with a higher dynamic range.

2.1.3 Creating HDR images

Since a digital camera has a limited range, multiple images of the same scene, at different exposures, are needed to create one HDR image (Cerman [3], DebevecFerwerda and Malik [4], Grossberg and Nayar [7]). A large range can be captured by combining a number of small ranges. Each image contains vital information - the overexposed images provide detail to the darker regions and the underexposed images provide detail to the brighter regions (McCollough [14]). By combining the useful information from each image, a composite image is created with detail in all areas. This process is done automatically in programs such as Adobe Photoshop and Photomatix. Although the image can now be stored, there are still problems when taking the images to make up the HDR image, as well as viewing it.

2.1.4 Problems with HDR images

Problems with HDR images fall under two categories:

1. Creating the HDR image

2. Viewing the HDR image

2.1.4.1 Problems with creating an HDR image

One major issue with creating a single HDR image is the need for multiple images. This means that each image needs to capture a consistent view of the same scene, with only exposure changing. Camera movement and subject movement are immediate issues, both of which produce ghosting [26]. This can be avoided by interpolating colour [21] or having multiple image sensors [15]. Having multiple image sensors is expensive because multiple image detectors are required, as well as having to align the optics required to take the image. Because of this problem, most HDR images are of still life, with very little movement by any subjects in the scene, and a tripod is generally used.

2.1.4.2 Problems with viewing an HDR image

Once the problems of creating an HDR image are overcome, there are still problems with viewing it. A typical monitor, and especially print media, is unable to display the dynamic range of the HDR image because both displays and print media do not have a large enough dynamic range (Cerman [3], Johnson and Fairchild [9], Kuang et al. [10], Ledda et al. [12, 11], McCollough [14], Nayar and Mitsunaga [15], Rempel et al. [18], Seetzen et al. [20, 19], Trentacoste [23], Ward [24, 26], Ward et al. [27, 28], Yuan et al. [30]). Some form of tonemapping is required to display an HDR image on a monitor or to print it. This process is discussed in Section 2.2.

2.2 Tonemapping

2.2.1 What is tonemapping?

Tonemapping is the process of mapping an HDR image to a Low Dynamic Range (LDR) image. This is used to overcome the problem of displaying HDR images. Various tonemapping algorithms exist (Cerman [3], Durand et al. [6], Grossberg and Nayar [7]) and there are even algorithms that try to do the reverse, replicating an HDR image from an LDR image (Rempel et al. [18]). An in depth comparison of tonemapping has been done by Kuang et al. [10]. Global tonemapping techniques (which applies the same transformation to every pixel of the image) and local tonemapping techniques (which apply a different transformation to different pixels, depending on their relative position - e.g. a bright pixel surrounded by dark pixels is made brighter, to accentuate the contrast).

An interesting part of the paper was the use of the Thurstone's Law, Case V (Thurston [22]), which relies on people judging two images at a time to create a scale on which images are best.

2.2.2 Problems with tonemapping

As much as tonemapping might seem like a solution to the issue of display (both on monitors and in print), detail from the original image is lost and the experience not fully replicated (Ledda et al. [12], Reinhard et al. [17]). The much used example is that of a car approaching you at night, with bright headlights, which appears very differently when experiencing this in real life, as when viewing a photograph of the event (Ledda et al. [11], Seetzen et al. [20], Trentacoste [23]). Tonemapping does not do the original scene justice, leaving it static with merely overexposed highlights.

Therefore, research has been undertaken to display HDR images with a dynamic range closer to that of the original image, as is discussed in the following section. However, we have found no research published on printing HDR images with full dynamic range, other than as an LDR image which has been created by tonemapping an HDR image. Existing printing techniques will be discussed in Section 2.4.

2.3 Displaying HDR images

Various techniques have been devised to view an HDR image on an electronic display. Four examples are discussed here:

- Fast Bilateral Filtering (Durand and Dorsey [5])
- Projector-based display (Seetzen et al. [20])
- LED-based display (Seetzen et al. [20, 19])
- Commercially available HDR display

2.3.1 Fast Bilateral Filtering

Durand and Dorsey [5] came up with a method of two-scale decomposition containing: a base layer, which records large variation and has the contrast reduced; and a detail layer. The colour layer is treated separately. How well this worked is inconclusive as the paper focuses more on the time taken to process the image, than the effectiveness of the algorithm.

2.3.2 Projector-based display

A similar model implemented in hardware is used by Seetzen et al. [20] in their projector-based display. This display has an LCD with a highly detailed image and a projector that projects onto the back of the LCD. The LCD panel is stripped of its own back-lighting and a Fresnel lens is placed behind the panel and all components are carefully aligned. According to Seetzen et al. [20], this display works reasonably well, but to achieve 10 000 cd / m^2, on a 15" screen would require 10 000 lumens - a huge amount, resulting from the inefficiency of the system.

2.3.3 LED-based display

Another method of fully reproducing the dynamic range of an HDR image on a display is suggested by Seetzen et al. [20, 19]. This design uses an LCD, stripped down as above, and a low resolution matrix of Light Emitting Diodes (LED) behind the LCD. The LEDs are brighter in the portion of the image that represents more light. One distinct advantage of the LED-based system over the projector-based display, is that it is considerably smaller, requiring space only for the LCD and LEDs, not a projector. As a sign of the success of this method, it is used in commercially available displays, discussed next.

2.3.4 Commercially available HDR display

There is an HDR display that is commercially available - the Dolby Brightside DR37-P (Dol [1]). This display uses a similar technology to the LED-based display mentioned above (Section 2.3.3) but is not that available since it has to be ordered directly from Brightside, and for a huge cost.

2.4 Printing Techniques

As mentioned above, we have found no published research on printing HDR images. Whereas displays have been created to view HDR images with a higher dynamic range, with some success, printing HDR images invariably requires tonemapping (Reinhard and Debever [16]).

2.5 Conclusion

High Dynamic Range images create a world of opportunity to capture a scene more accurately. However, displaying these images on an electronic display require huge amounts of effort, either by reducing the dynamic range and compromising the accuracy, or by creating elaborate display systems, each with its own weaknesses. Printing HDR images is limited to reducing the dynamic range and the original scene cannot be reproduced at this point in time.

Chapter 3

Method

3.1 Introduction

To try and retain the dynamic range of an image when printing, we propose a method of printing on multiple transparencies. We choose this model over other models such as the elaborate process of shining light from different angles, and printing layers of ink onto one sheet of paper, because it seems the most feasible, and most reasonable. It also correlates well with the process of creating an HDR image, as discussed in Chapter 2. Our model replicates dark areas of the original HDR image by printing ink on more of the transparencies than bright areas, which will have ink printed on fewer transparencies. In doing this we anticipate that each additional transparency will add at least one extra bit of dynamic range. Therefore we hypothesize that this process will overcome the limited dynamic range of the printer. Coupled with the multiple transparencies, we have in place a bright light source behind the transparencies. As with the number of transparencies, we expect the brightness of the light source to affect the resulting dynamic range. This is because, by our proposed model, a dark area of an image will produce transparencies with a lot of ink, blocking out a lot of light, and a light area will produce transparencies with little ink, letting a lot of the light through. Therefore a greater dynamic range should be perceived, because of a greater difference between light and dark.

To obtain the exact method of conversion from original HDR image to set of transparencies, a number of experiments have been devised. These involve calibrating the printer and camera, so a relationship can be formed between the value in the image file, with the perceived luminosity when placed in front of a bright light source (Section

3.2), as well as creating the program to use this relationship (Section 3.4). Firstly, to calibrate the camera, we take a series of luminosity readings of grid patterns. Secondly, we take a series of luminosity readings of shades of grey. These two experiments we use to create the relationship between the pixel value and luminosity value, but before the experiment are unsure what this relationship will look like. With the data from these experiments, we then derive a formula, to apply on a pixel-by-pixel basis to create the transparencies (Figure 3.1). Thirdly, we take luminosity readings on two layered transparencies, with shades of grey, to observe the effect of layering shades of grey, critical to our proposed method. We use this to help decide on the exact method of conversion from HDR to transparencies (Section 3.4).

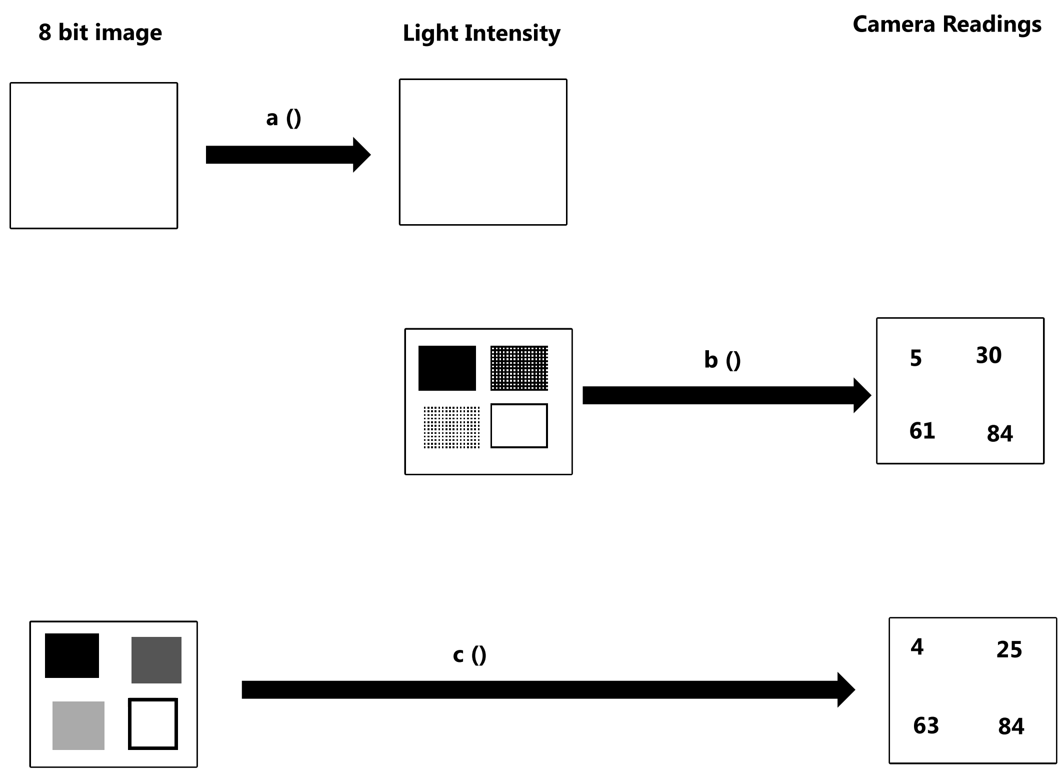

Figure 3.1: Overview of experiments

3.2 Calibration

Before printing the HDR image to transparency, we first of all calibrate the response of the printer, to give the desired luminosity based on the luminosity value of the original image.

3.2.1 Building and Operating a Light Meter

We perform all experiments using a webcam (Logitech QuickCam© Sphere™ AF) as a light meter, turning as many automatic settings off as possible to minimize experimental error. All settings manually set appear in Table 3.1. It is important to note that we have the settings for hue, saturation, sharpness, white balance, gamma and backlight compensation all set to the lowest possible settings to minimise any curves being applied to the image rendering process. We then set the remaining settings to give distinguishable readings across the entire range of transparencies. This is done by placing the lightest and darkest transparencies on the light source, an overhead projector, and adjusting the gain and exposure until the difference between lightest and darkest, when taking a photo of the transparencies, is the greatest.

Setting	Value
Exposure	1/20
Gain	980
Brightness	2000
Contrast	784
Hue	0
Saturation	0
Sharpness	0
White Balance	0
Gamma	1000
Backlight Comp	0

Table 3.1: Webcam settings

On the webcam itself, we cover the red "On" light to avoid interference, so it was not reflected by the transparency, creating additional light. We also fit a cylindrical, black tube approximately 10cm in length, to block out ambient light, around the webcam so accurate readings can be taken without the influence of ambient light. We place seven plain sheets of A4 paper between the light source and the transparency so readings can be taken without being saturated. We found that this number of sheets of paper blocked out enough light, but still allowed enough through to get workable readings across the entire range of luminosity. To use the webcam as a light meter, we use Adobe Photoshop CS3 Extended and, import from the Logitech camera (File -> Import), repeating this procedure for each run of the experiment. For each image captured, we manually select the critical portion (see Figure 3.2) of the image because the image captured contains portions of the inside of the cylindrical tube.

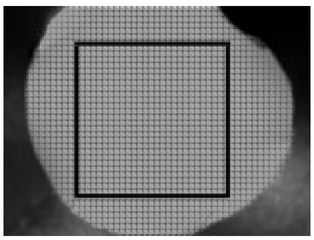

Figure 3.2: Raw image from webcam, with critical section selected with the marquee tool

We now proceed to retrieve a luminosity reading of each image. Having defined an action to crop, perform a Gaussian blur (250 pixel radius) five times in succession, convert to greyscale and perform another Gaussian blur (250 pixel radius) to an image, we apply this action as a batch process to produce a uniform, grey image for each of the original photos (see Figure 3.3). We now use the red, green or blue value (all equal now since the image is greyscale) as our luminosity reading. This number represents the amount of light received, and hence luminosity, and so in our luminosity scale, a low number represents a small amount of light, and a higher number a larger amount of light.

Figure 3.3: The original image is cropped and blurred to get a luminosity reading

3.2.2 Experiments

As described in Section 3.1, we wish to derive a function, $a()$ that relates the shade of grey in an image and the luminosity of that shade when printed on transparency and placed in front of a bright light source. This we can then use to relate the greyscale value in an HDR file to a greyscale value to print, to produce a desired luminosity (which should represent the luminosity in the HDR file). To derive this function, a(), we perform two experiments. Firstly, Experiment 1 (Grid Patterns), to derive the function b() between a known percentage of ink and luminosity. Secondly, Experiment 2 (Shades of Grey), to derive a function c(), between luminosity and shades of grey. These functions can then be combined to derive the required function a() (see Figure 3.1).

3.2.2.1 Experiment 1: Grid Patterns

In Experiment 1 (Grid Patterns) we use grid patterns so that the amount of black on the transparency is known (see Figure 3.4). Using patterns based on a four-by-four pixel base, a region with, for example, 50% black is easily created using a two-in-four pixel pattern. Using the process described above, we can populate a table and create a regression curve b(), based on the average of three runs of the experiment, shown in Figure 3.5.

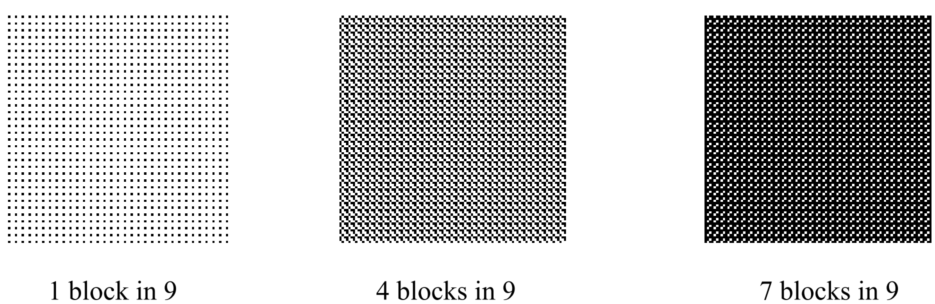

Figure 3.4: Examples of some of the grid patterns used in calibration

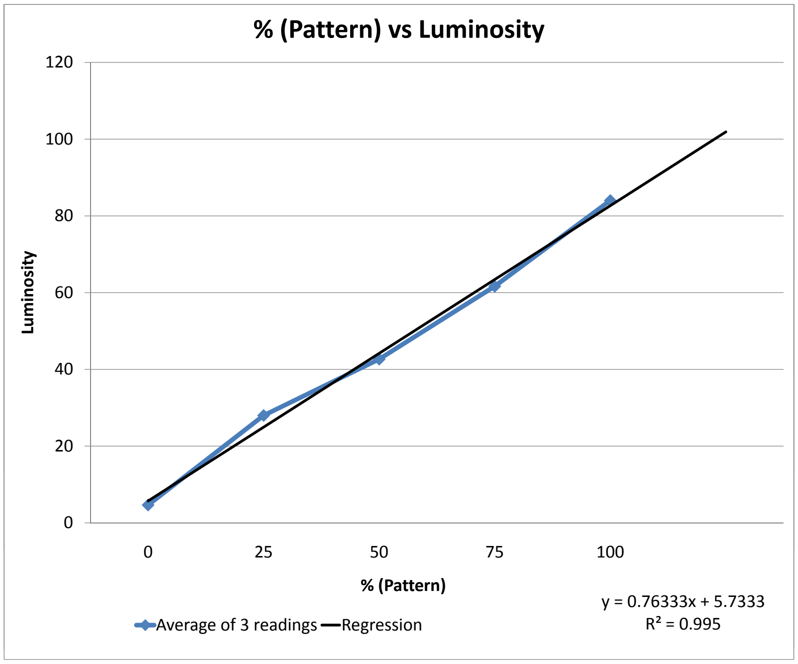

Figure 3.5: Pattern vs Luminosity

3.2.2.2 Experiment 2: Shades of Grey

Now, having calibrated the camera with the grid patterns, we perform Experiment 2 (Shades of Grey) three times to find the relationship between shade of grey and luminosity. We do this by printing a shade of grey – the shade represented by a number between 0 and 255 (inclusive) (see Figure 3.6) and measuring the luminosity in the same manner described above (Section 3.2.1). This results in populating another table, the values of which we use to successfully derive c(), the relationship between shade of grey and luminosity, shown in Figure 3.7.

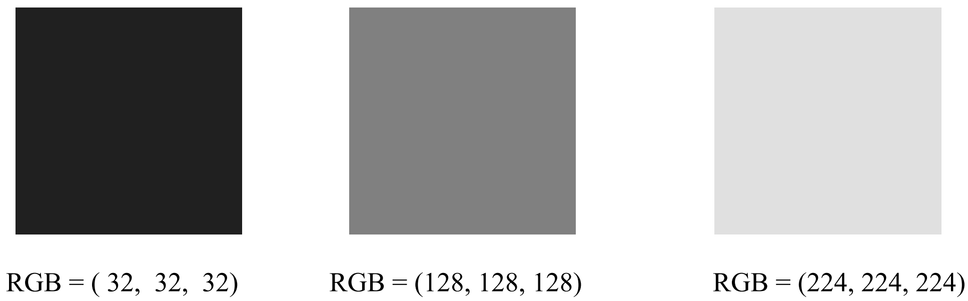

Figure 3.6: Examples of some of the shades of grey used in calibration

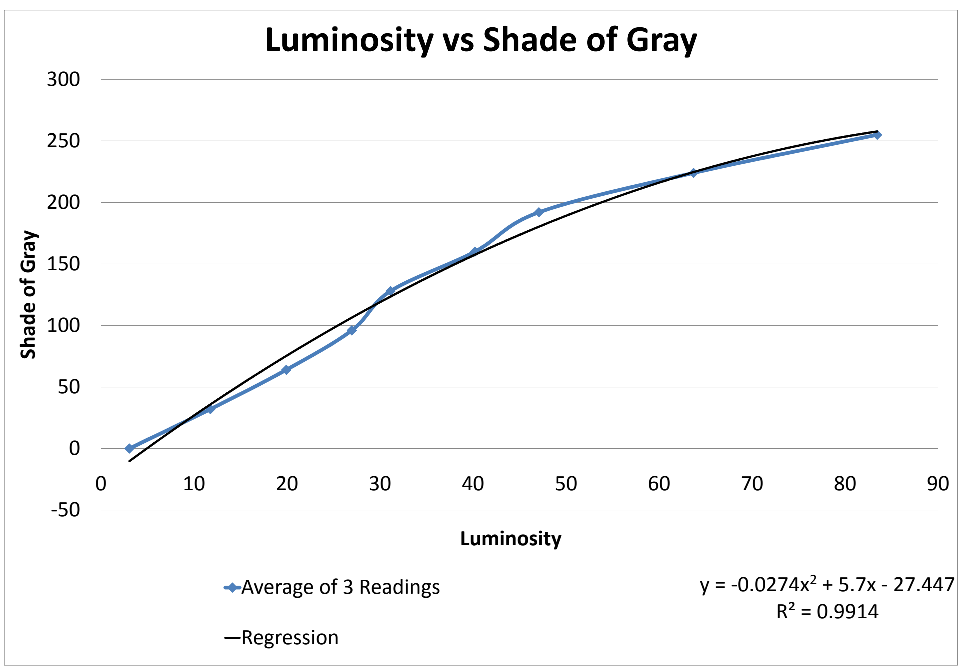

Figure 3.7: Luminosity vs Shades of Grey

Combining functions b() and c(), we derive function a() (see Figure 3.1) to convert an HDR image into an 8 bit images, described in more detail in (Section 3.4).

3.2.2.3 Experiment 3: Layering

To determine the effect of layering transparencies, we perform Experiment 3 (Layering), and record the luminosity when two transparencies with shades of grey are placed on top of each other. This experiment uses the same transparencies used in Experiment 2 (Shades of Grey) (Section 3.2.2.2), this time using two layered, as opposed to just one transparency. We noticed some extremely revealing results (see Table 3.2). Most notable is the luminosity recorded when a transparency with 50% grey (represented by 128), is placed on top of another transparency with 50% grey. What we notice when performing the experiment is that aliasing is visible when the transparencies are layered (see Figure 3.8). This immediately makes us examine the print more closely and what we notice is that instead of printing a smooth, continuous grey, the printer actually prints a pattern.

	0	**32**	**64**	**96**	**128**	**160**	**192**	**224**	**255**
0	10	14	21	24	24	33	32	50	61
32	14	15	25	33	45	43	51	66	79
64	21	27	40	49	51	63	68	87	123
96	25	34	43	39	28	66	81	120	144
128	25	42	46	73	56	73	90	129	147
160	33	49	60	65	76	82	93	150	153
192	39	59	82	72	83	118	146	152	153
224	47	68	120	125	130	151	151	152	152
255	64	79	122	141	152	152	149	152	152

Table 3.2: Layering of two transparencies, 0 = completely black, 255 = fully transparent, and their corresponding luminosity, as recorded by the light meter

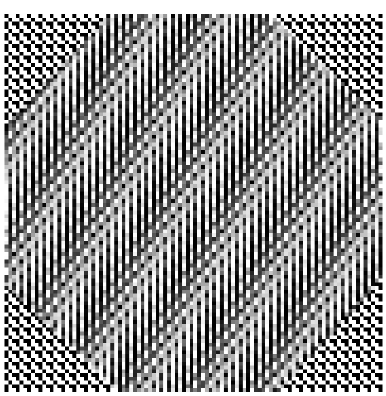

Figure 3.8: Aliasing was observed with two transparencies layered on top of each other

Having related the light intensities captured by the camera to quantities on ink produced by the printer, by calibrating the response of the black and white printer and building and using a light meter to measure the printer response, we proceed to repeating the process for colour (Section 3.3), before devising and testing three techniques for converting HDR to transparency (Section 3.4).

3.3 Extension to Colour

To extend the model to colour, each of the three experiments mentioned in Section 3.2.2, was repeated for each of the red, green and blue channels i.e. for each experiment, wherever a shade of grey was concerned, the experiment would be done with a shade of red, then repeated with a shade of green, and finally repeated with a shade of blue. In the colour version of Experiment 1 (Grid Patterns) (Section 3.2.2.1), for the red channel, the grid patterns had full red (RGB value = (255, 0, 0)), instead of full black (and similarly for green, and blue). The results of Experiment 1 (Grid Patterns) for each of red (Figure 3.9), green (Figure 3.10) and blue (3.11) are shown below. These experiments resulted in the following formulas for b() (Figure 3.1), calculated by regression from the average of three runs of each experiment:

Red: y = -0.5896x^2+ 0.001149x + 122.4

Green: y = -0.6322x^2 + 0.001494x + 127.0

Blue: y = -0.8420x^2 + 0.001764x +124.0

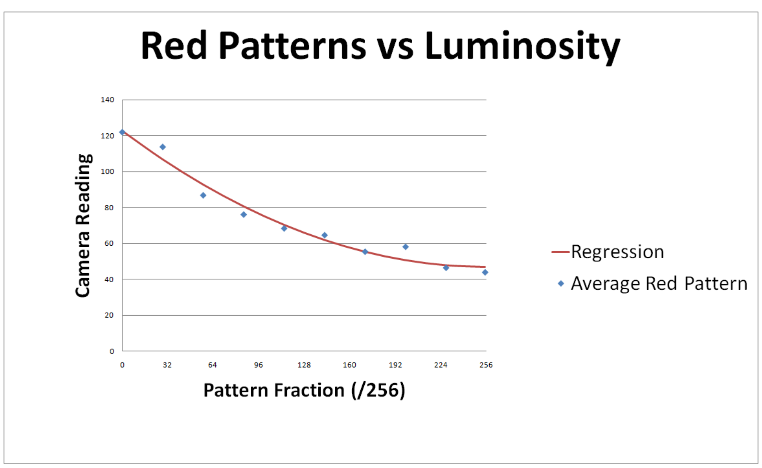

Figure 3.9: Red Pattern vs Luminosity

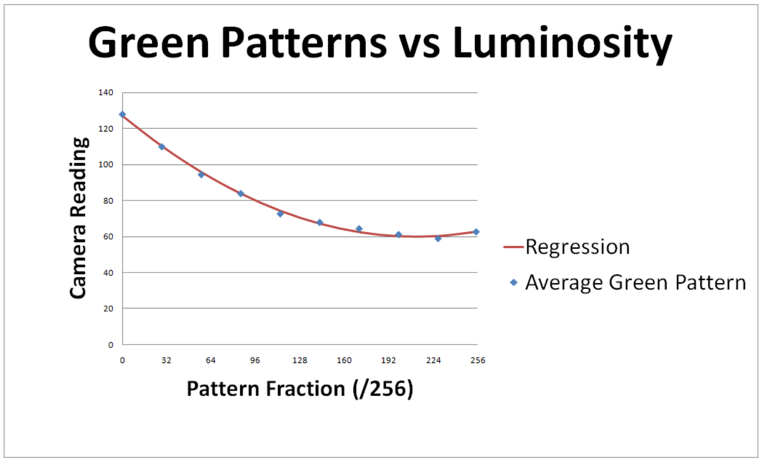

Figure 3.10: Green Pattern vs Luminosity

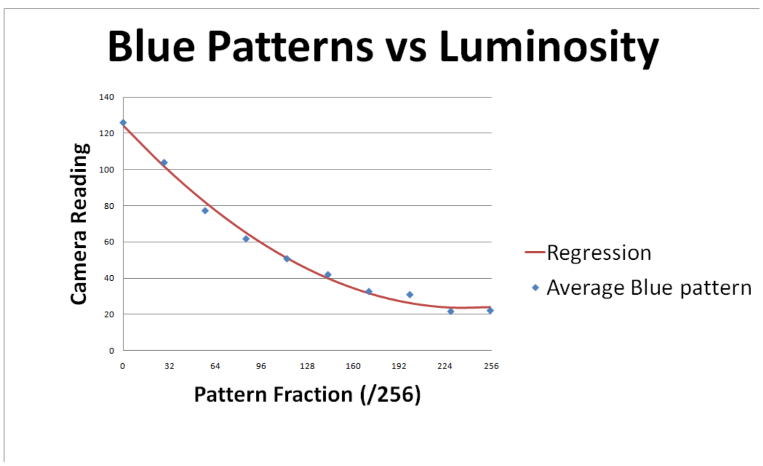

Figure 3.11: Blue Pattern vs Luminosity

In Experiment 2 (Shades of Grey), for the red channel, was done with red (RGB value = (255, 0, 0)) to transparent (RGB value = (255, 255, 255)) as equivalent of black to transparent range. Therefore, in Experiment 2 (Shades of Grey), a grey value of 128 has an equivalent RGB value, for the red channel, of (255, 128, 128). This translates to a shade of red halfway between full red, and fully transparent. Again, similar changes were made for the green and blue channel experiments. The results of Experiment 2 (Shades) for each of red (Figure 3.12), green (Figure 3.13) and blue (3.14) are shown below. These experiments resulted in the following formulas for the inverse of c() (Figure 3.1), calculated by regression from the average of three runs of each experiment:

Red: y = -0.8015x^2+ 0.001729x + 135.8

Green: y = -0.5654x^2 + 0.001216x + 129.5

Blue: y = -1.077x^2 + 0.002554x +133.6

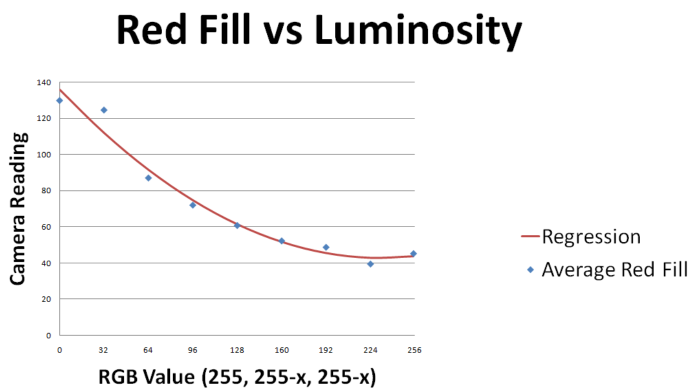

Figure 3.12: Red Fill vs Luminosity

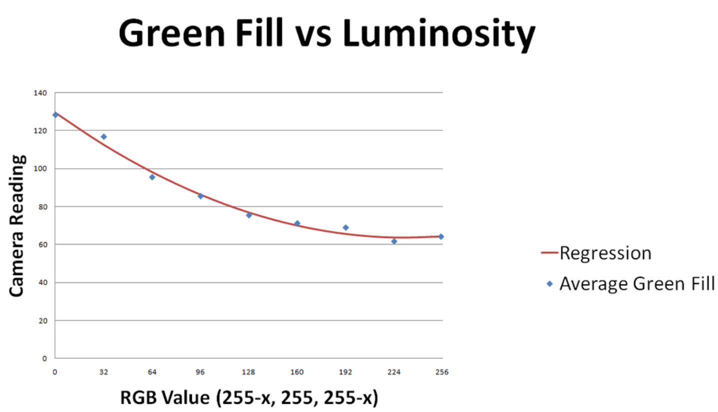

Figure 3.13: Green Fill vs Luminosity

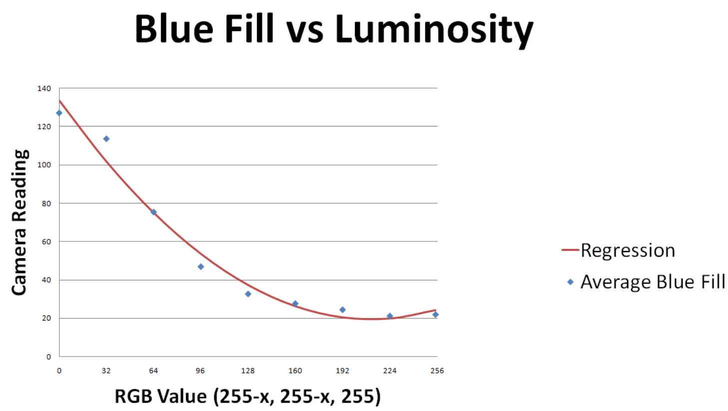

Figure 3.14: Blue Fill vs Luminosity

As one can see, these experiments show very similar functions, and when combining b() and c() for each of the colour channels, we get an equation for a(): y $= x$ (Figure 3.1). Therefore, unlike the black and white printing, the colour printer is 'well behaved' and no conversion needs to be made before printing i.e. given a value in the original image, no function needs to be applied to get the correctly perceived luminosity. With the calibration for both colour and black and white, we can devise methods for converting HDR to transparency.

3.4 Converting HDR to Transparency

We now derive a method of converting an HDR image to multiple 8 bit files, which can be printed on multiple transparencies, using function a() obtained in Section 3.2. We show three techniques, each derived due to an observation in the previous one, but full evaluation will be left for Chapter 4. The same HDR image was used for all techniques, saved in OpenEXR format - .EXR. To apply the principles proposed in each technique, a C++ program is made, using the open source OpenEXR libraries, available at http://www.openexr.com. Each pixel is scanned in, the given technique applied to it, and the given number of transparencies are produced in portable pixmap (PPM) format. The reason for this format is its ease of creation, and cross-platform portability. Since efficiency is not a major concern, but rather implementation of the principles conceived, we refer throughout to the actual principles, and not to specific code, which is just the tool used to implement the principles.

3.4.1 Technique 1

We will first consider the case of black and white. Because of aliasing when layering shades of grey, observed in Experiment 3 (Layering) (Section 3.2.2.3), we decide to use an approach that doesn't use a shade of grey on top of another shade of grey in Technique 1. (Note here, that in this instance, shade of grey does not include the darkest shade of grey i.e. black.) Therefore, we wish to create a system to take an HDR image as input, and produce a given number, say n, of transparencies. Because of the aliasing observed in Experiment 3 (Layering) (Section 3.2.2.3), we are unsure as to what the effect of layering transparencies is, and therefore assume that the light is attenuated in an adding fashion (to be explained now, and to be evaluated in Chapter 4). The explanation that follows deals with one pixel of the HDR image, and the correlating pixel on each transparency (e.g. the pixel at (x,y) coordinates (0, 0)). Given an HDR input, we wish to produce n transparencies, such that together they simulate the luminosity of the HDR input, as best as it can. But, only one transparency can have a shade of grey, the other (n - 1) transparencies must each be either fully black, or fully transparent, so as to avoid aliasing. This is represented pictorially in Figure 3.15, for 4 transparencies.

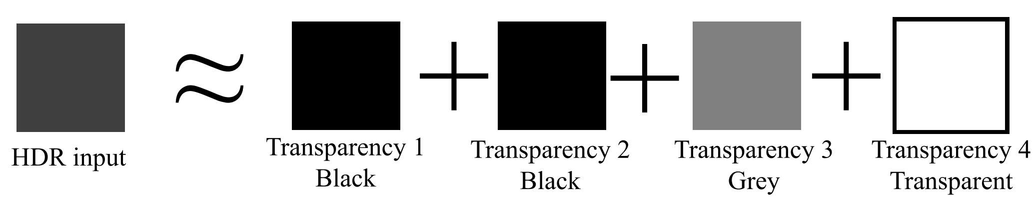

Figure 3.15: Given a pixel of HDR input, n transparencies are produced, with only one transparency a shade of grey to avoid aliasing

To determine how many black and how many transparent layers, as well as the shade of gray to use on the shaded transparency, the following algorithm is used for each pixel:

- Given n = number of transparencies, lum = luminance of pixel in HDR file (say a value between 0 and 1)
- Loop i = 0 to n

  - Let m = lum modulus (1/n) (i.e. remainder when lum is divided by (1/n))
  - Let tnum = lum div (1/n) (i.e. the value of lum divided by (1/n), throwing away the remainder)
  - IF i < tnum
    * Then this should be transparent, i.e. we haven't got to the grey layer yet (we're on the far right of Figure 3.15)
  - IF i equals tnum
    * Then we are on the grey layer, and the actual colour to be printed is based on the function a(), described in Section 3.1 and derived in Section 3.2.2, and mapped to a value between 0 and 255.
  - ELSE
    * The pixel should be black (we're on the left of the grey pixel in Figure 3.15)

We apply the same principle to a colour image, dealing with each channel separately (the shades of grey and black converted as discussed in Section 3.3). In the case of colour, the function a() is just y = x, so the colour is simply mapped linearly to a value between 0 and 255 if i equals tnum. However, although we believe this to increase the dynamic range, since each colour channel can now be one of more than 256 colours (details in Chapter 4), when produced and viewed with a bright backlight, the dark areas appeared too dark, and missing detail. We therefore believe our assumption of simply adding the attenuation of light for each transparency to be wrong, and consider a new technique (Section 3.4.2), based more rigourously on the idea of applying a function on the incoming intensity of light, to get the perceived intensity of light.

3.4.2 Technique 2

In this technique, and the next (Section 3.4.3), we base all our experiments on colour images. This is because we observe no aliasing when layering multiple colour transparencies, printed on the 'well behaved' colour printer (Section 3.3). Therefore we can be more creative in devising techniques for converting HDR to transparency and use values between 0 and 255 on any number of transparencies. Since Technique 1 (Section 3.4.1) didn't recreate the dynamic range correctly, Technique 2 proposes a method in which the $Intensity_{out}$ is related to $Intensity_{in}$ in the following way:

$$Intensity_{out} = Intensity_{in}(1 - k(1 - \alpha)) \tag{3.1}$$

where α is a constant based on the brightness of the backlight and k is the amount of ink.

Graphically, this is shown in Figure 3.16.

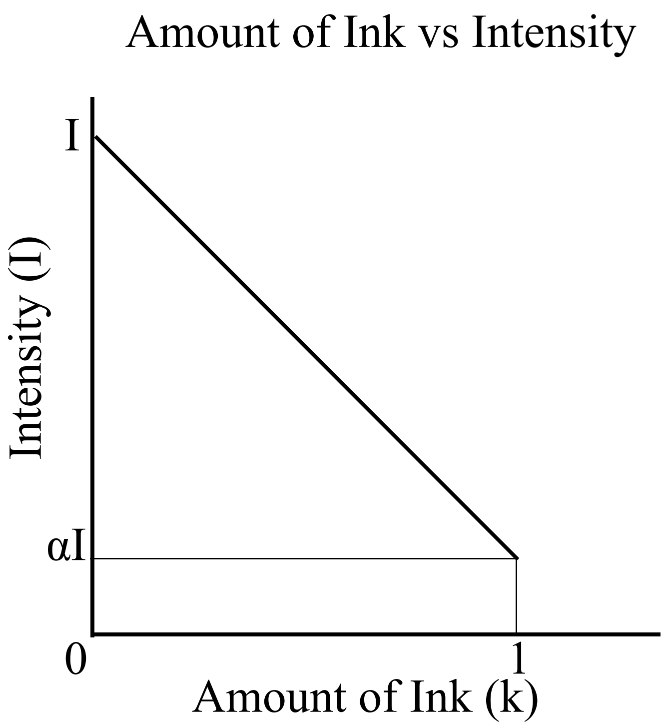

Figure 3.16: Amount of Ink vs Intensity

The formulation of Equation 3.5 is derived from a linear interpolation of the amount of ink, k (ranging from 0 to 1) and Intensity, I (ranging from αI to I). Equation 3.5 represents the effect of one transparency. To simulate the effect of two transparencies, $Intensity_{in}$ for the second transparency is $Intensity_{out}$ from the first transparency. In other words, Equation 3.5 becomes:

$$Intensity_{out} = Intensity_{in}(1 - k_1(1 - \alpha))(1 - k_2(1 - \alpha)) \tag{3.2}$$

where k_m is the amount of ink on transparency m.

Generalising this for n transparencies, Equation 3.5 becomes:

$$Intensity_{out} = Intensity_{in}(1 - k_1(1 - \alpha))(1 - k_2(1 - \alpha))...(1 - k_n(1 - \alpha)) \qquad (3.3)$$

where n = number of transparencies.

To solve Equation 3.5 analytically, there must be just one unknown, since there is just one equation. To achieve this, we set:

$$k_1 = k_2 = ... = k_n \qquad (3.4)$$

i.e. set the amount of ink to be the same on each transparency.

Now, substituting Equation 3.4 into Equation 3.5, and calling the amount of ink k:

$$Intensity_{out} = Intensity_{in}(1 - k(1 - \alpha))^n \qquad (3.5)$$

Solving for k is then a simple case of reorganising and is:

$$k = \frac{1 - \sqrt[n]{lum}}{1 - \alpha} \qquad (3.6)$$

The algorithm to follow to create convert an HDR image to transparencies is now completely different to the one we devised in Section 3.4.1. Firstly, there is no need to loop through each transparency, since they're all the same, so one PPM file will be output. Secondly, the colour, b, to print needs to be derived from k, the amount of ink. This is a simple linear mapping given by:

$$b = 255(1 - k) \qquad (3.7)$$

where b is the colour, in the range 0-255, to print.

Therefore, the colour to print is easily found, by applying first Equation 3.6, then Equation 3.7, to each pixel. Although this now represents the dynamic range more evenly from dark to light, making the scene appear closer to the original HDR, it is apparent that even though the darkest area is darker, and the lighter area lighter, the dynamic range is not increased. This is because each pixel can still only be one of 256 colours (discussed in more detail in Chapter 4). Therefore, we modify the solution of Equation 3.3, to be able to print different amounts of ink on each transparency, and derive our final technique, Technique 3 (Section 3.4.3).

3.4.3 Technique 3

As discussed in Technique 2 (Section 3.4.2), we need to devise a new method of solving Equation 3.3. Instead of setting k, the amount on each transparency, to be equal, and solving analytically, we use numerical methods to solve Equation 3.3. Since b (in Equation 3.7), the colour to be printed on transparency, ranges from 0 to 255 we work in a 'base 256' system. What this means we explain in the following algorithm:

As in Technique 1 (Section 3.4.1), we have: n = number of transparencies; lum = luminance of pixel in HDR file (say a value between 0 and 1). And, as before, we do this algorithm for each pixel.

- Set up array b = $(b_1, b_2, ..., b_n)$ = (0, 0, ... , 0), to represent the colour on each transparency
- Substitute this into Equation 3.3, solving for each $k_i = 1 - \frac{b_i}{255}; (1 \leqslant i \leq n)$ (from Equation 3.7)
- IF this value (the perceived output with the given amount of ink on each transparency, as represented by b) is close enough to lum*
  - Then the values in b represent the correct amounts of ink, and the values for this pixel are calculated and we can move onto the next pixel
- ELSE
  - b does not represent the correct values, and we must 'add one' to b, and this is evaluated, such that all values are tried until the correct combination is found
  - i.e. b starts as (0, 0, ..., 0, 0), then, after the 'add one', should become (0, 0, ... , 0, 1) etc. until (0, 0, ..., 0, 255), at which point an 'add one' changes b to (0, 0, ... , 1, 0). This is representative of a 'base 256' system.
- This will try every possible combination of values for b, until the correct combination (i.e. the correct preceived luminance) is found
- * Since there are 256^n possible combinations for b, we say that the perceived luminance is "close enough to lum" when the absolute difference is less than or equal to $\frac{1}{256^n}$.

3.5 Conclusion

We now believe to have the solution to the problem, reproducing the dynamic range of an HDR image on print media, and set out to do a detailed evaluation in Chapter 4 on each of the techniques described in Section 3.4.

Chapter 4

Evaluation

4.1 Introduction

To evaluate the dynamic range of the printed image, it is necessary to be able to measure the dynamic range. The first technique we devised to measure dynamic range is to create an HDR image by taking photos of the printed image, and to compare this to the original HDR image. This, however, has its limitations because this image will have a high dynamic range, regardless of whether the printed image if high or low dynamic range. The reason for this, is that there will be slight variations of colour of even just one pixel on one transparency. One reason is due to artefacts when printing the ink onto the transparency - the printer may print one pixel slightly different shades, even though it represents one colour. As well as this, the transparency could be bent, even slightly, which would cause the ambient light to reflect, and the light from the backlight to refract. To try and overcome these problems, a number of runs could be done, taking photos at different times, and creating several HDRs. These could then be evaluated, looking at each pixel individually. This is not, however, a feasible solution, since the HDR image taken of the printed image would have to be aligned, so that each pixel on the HDR image maps to the correct pixel of the printed image. This would involve resizing, skewing and transforming which could still leave the pixels incorrectly mapped. To do such a process would not only be time consuming, to a very large degree, but could well be inexact, causing inaccurate results. Therefore we devised a second technique.

The second technique we devised, involves comparing the lightest and darkest areas of the printed image. Since the dynamic range of the final image is not just the difference between light and dark, but also the range of luminance in between, the techniques of printing are also evaluated mathematically, in some cases proving that the printed image cannot be representative of a high dynamic range, and in other cases, by the given model, can and will be.

As a second form of evaluation, the effect of the number of transparencies used (Section 4.3), will be discussed. Again, we use the second technique described above. The effect of two transparencies will be discussed in Section4.2, with a more general case of n transparencies in Section 4.3).

4.2 Dynamic Range

The first method we test is when an HDR image is tonemapped and printed onto a transparency. Regardless of the tonemapping technique used, the following will hold, since tonemapping is the conversion of HDR to 8 bits, as described in Section 2.2. The darkest any area can be is that of a black region. In this region the backlight will be blocked by one layer of black, resulting in luminosity values the same as those we recorded when calibrating for full black, in Section 3.2. The brightest any area can be will be transparent, and will not block any light from the backlight. Again, this will result in the same values we recorded for full transparency, in Section 3.2. As pointed out in Section 2.2, the dynamic range is now limited to what can be stored in 8 bits per colour channel. So although brighter areas may appear brighter, and darker areas darker, the range of colours has remained at 8 bits worth, per colour channel, and the resulting dynamic range has not increased. Therefore, although perhaps more perceptually pleasing (elaborated on in Section 4.4), this is not a solution to increase dynamic range of a printed HDR image. For two transparencies, since this system is designed for one output, we consider the case of duplicating the single transparency. This results in the darkest area being the combination of two black transparencies, and the lightest that of two transparent transparencies. However, since it is one image duplicated, the range remains at 8 bits per colour channel because each pixel can be 1 of 256 colours.

The second method we test is Technique 1 (Section 3.4.1), in which, for a given pixel, just one layer can be shaded, each of the rest either fully black, or fully transparent. Looking at this method for one transparency, the same can be said as with printing a tonemapped image. The darkest an area can be is fully black, since, following the algorithm for an input value of 0 (black), the output will be 0 (black) And the lightest an area can be is fully transparent, since, following the algorithm, an input value of 1 (white) gives an output of 255 (transparent). Therefore, with just one transparency, this technique does not improve the dynamic range either. For two transparencies, the darkest an area could be is with both transparencies black. Following the algorithm, an input value of 0 (black) results in an output value of 0 (black) on both transparencies. The brightest area, on the other hand, can again be fully transparent. If the input value is 1 (white), the algorithm gives an output of 255 (transparent) on both transparencies. Combinatorially, the algorithm gives 511 colours per colour channel. This is because the one transparency can be fully black (0), while the other can range through all 256 shades (0-255 inclusive), or the one transparency could be fully transparent (255), with the other ranging from 1-255 inclusive. Note that order does not matter by this model, since it is assumed that the layers attenuation on the light is just added. Therefore the dynamic range has increased since the difference between the darkest and lightest areas has increased, as well as the range of colours. The perceptual accuracy, as alluded to in Section 3.4.1, and partly causing the derivation of Technique 2 (Section 3.4.2) will be more fully elaborated upon in Section 4.4.The effect of adding more transparencies with Technique 1 (Section 3.4.1) will be discussed in Section 4.3.

The third method we test is Technique 2 (Section 3.4.2). Using the algorithm in this model, we again see that, with one transparency, a range of 0 (black) on the transparency to 255 (transparent) on the transparency can be achieved. With two transparencies, the algorithm produces the same lowest (black on both transparencies) and highest (transparent on both transparencies) values as Technique 1. However, since each transparency produced is the same, the range is the same as that of one transparency. So, although the darkest area will be darker, and the lighter area lighter by this model, the number of colours per colour channel remains at 256.

The fourth and final method we test is Technique 3 (Section 3.4.3). With one transparency the model is identical to Technique 2 as Equation 3.3 will merely become Equation 3.1, which is identical to 3.5 with $n = 1$. Therefore the same conclusions can be drawn about dynamic range. With two transparencies, however, the model becomes very different. In terms of lightest and darkest areas, it is identical again to Technique 2, as the inks on each transparency can both be set to give black (0 - same scenario as Technique 2) or both be set to give transparent (255 - again, same scenario as Technique 2). The major difference between Technique 2 and 3 lies in the range that can be produced. Since, by the algorithm, each transparency can be a different colour, and order doesn't matter as changing the order of $k_1, k_2, ..., k_n$ in Equation 3.3 does not change the output (multiplication is commutative for real numbers), the total number of colours per colour channel, calculated combinatorially, is $\frac{256^2+256}{2}$ = (257 choose 2) = 32896. This is clearly a much broader range, achieved with only two transparencies.

We now move on to seeing the effect of the number of transparencies for tonemapping and each of Techniques 1, 2 and 3.

4.3 Effect of Number of Transparencies

Having looked at the case of one and two transparencies, we now consider the general case of n transparencies.

For all 4 models discussed, tonemapping and the 3 original techniques, since the darkest area produced is black, and all models have the capability of every single transparency being black, the darkest area produced is that of n black transparencies, in all four cases. Although not tested through rigourous experiment, we oberve that more black transparencies layered with a light source behind, blocks out more light, creating a perceptually darker area. Since we have not tested this, we can not know the exact nature of this function, although we predict a linear function, as described for Technique 2 (Section 3.4.2) and Technique 3, in Section 3.4.3 to be more accurate than a simple additive function, as described for Technique 1, in Section 3.4.1. This is an area which could be expanded upon, perhaps creating more accurate models, based on similar principles.

As with the darkest area, all four techniques can produce the same upper limit, for the lightest area. All n transparencies can be fully transparent, leaving the luminosity as just the backlight. Therefore the upper limit of luminosity is directly proportional, and equal, to the luminosity of the backlight.

The four techniques, do not, however, create the same range between theses upper and lower limits. The number of intensities, and distribution of these intensities, varies.

Tonemapping produces the same number of intensities, regardless of the number of transparencies. This is because each transparency is identical and the range limited to the range on any given transparency. Hence the number of transparencies will only affect the darkest point, and the range will still be limited to the printer's range - each pixel can be one of 256 colours per colour channel, regardless of what function is applied to calculate perceived luminosity. Therefore, tonemapping has the following relationship:

$$Range\,of\;Intensities = 256 \tag{4.1}$$

Technique 1, when worked out combinatorially, comes out at:

$$Range\,of\;intensities = 256 + (n-1)255 = 255n + 1 \tag{4.2}$$

which is a linear scale. In essence, each transparency gives an extra 256 values of range per colour channel. This is what we expected with the additive model used in Technique 1.

Technique 2 suffers from the same shortcomings as tonemapping, as each transparency is the same. Therefore it too is limited to the printer's range, regardless of the number of transparencies. It therefore also has the constant relationship expressed in Equation 4.1.

We expect Technique 3 to increase dramatically with each extra transparency, as it is based multiplicative function basis, as opposed to the additive nature of Technique 1 (or constant nature of tonemapping and Technique 2). When calculated combinatorially, Technique 3 has the following values for n = 1, 2, 3:

n = 1: Range of intensities = 256

n = 2: Range of intensities = 32896

n = 3: Range of intensities = 2828056

Here, the number of transparencies has a huge effect on the dynamic range reproducible.

4.4 Perceptual Accuracy

Since a detailed analysis was not done as to exactly how well the final image recreates the scene for every pixel, but more on the recreation of amount of dynamic range, we will deal very informally with perceptual accuracy here. This is another area which could be researched further, correlating directly with research proposed into deriving the function that attenuates light, discussed in Section 4.3. We observe the tonemapped image to appear pretty, for lack of a better word, since, although not creating the correct luminances, it try to use the detail from every pixel, giving an idea of the colour. This does not work well when multiple transparencies are printed and viewed on top of each other, as the incorrect luminance seems accentuated. Technique 1 produces a better idea of the luminance in the scene, but, as the additive model seems to leave the dark areas too dark, in comparison to the light areas, the effect is worsened with more transparencies. Technique 2 gives a great reproduction of luminances, with the dark areas appearing dark, and the light areas appearing light. When viewing the image though, we do notice the lack of range of luminances, so the luminance is correct only to a certain degree. Finally, we observed the results of Technique 3, expecting to feel immersed in a near real world experience. Unfortunately this is not the case, due to two reasons. Firstly, the algorithm used is extremely computationally expensive, so the image had to be downsampled to a lower resolution, and only a two transparency run could be completed in a realistic time. With the rapid increase in range of luminances comes the downside of having to check all the options, which is a long process. There are probably ways to improve the efficiency of the algorithm, offering a possibility of further research. The second reason is one that applies to all the techniques. And that is the difficulty in lining up the transparencies precisely, which may be impossible if warped slightly in the printing process. One way of overcoming this would be to slightly blur all but the top transparency. This would require more research into the effects of blurring and the adjustment of the model accordingly.

Chapter 5

Conclusion

5.1 Conclusions

In concluding this research we note, with some satisfaction, the goals achieved.

1. We are able to create HDR images (Section 2.1.3)

2. We are able to relate the light intensities captured by the camera to the quantities of ink produced by a printer by (Section 3.2):

   (a) Calibrating the response of printers across their entire range (Section 3.2.2)

   (b) Building and using a light meter to measure printer response (Section 3.2.1)

3. We are able to devise methods of deriving the dynamic range of a printed image (Chapter 4)

And most importantly, techniques have been devised to reproduce the dynamic range of an HDR image on printed medium, with success. We also conclude that the number of transparencies used affects the dynamic range of output. The effect of the number of transparencies on dynamic range is based on the technique used.

Bibliography

[1] High dynamic range displays. URL `http://www.dolby.com/promo/hdr/technology.html`. [Last Accessed: 1 June 2009].

[2] Aseem Agarwala, Mira Dontcheva, Maneesh Agrawala, Steven Drucker, Alex Colburn, Brian Curless, David Salesin, and Michael Cohen. Interactive digital photomontage. In *SIGGRAPH '04: ACM SIGGRAPH 2004 Papers*, pages 294–302, New York, NY, USA, 2004. ACM.

[3] Lukas Cerman. High dynamic range images from multiple exposures. Master's thesis, Czech Technical University in Prague, 2006.

[4] Paul E. DebevecFerwerda and Jitendra Malik. Recovering high dynamic range radiance maps from photographs. In *SIGGRAPH '97: Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 369–378, New York, NY, USA, 1997. ACM Press/Addison-Wesley Publishing Co. ISBN 0-89791-896-7.

[5] Frédo Durand and Julie Dorsey. Fast bilateral filtering for the display of high dynamic range images. In *SIGGRAPH '02: Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, pages 257–266, New York, NY, USA, 2002. ACM. ISBN 1-58113-521-1.

[6] Fredo Durand, Frédo Dur, and Julie Dorsey. Interactive tone mapping, 2000.

[7] Michael D. Grossberg and Shree K. Nayar. High dynamic range from multiple images: Which exposures to combine. In *Proc. ICCV Workshop on Color and Photometric Methods in Computer Vision (CPMCV)*, 2003.

[8] Jack Howard. *HDR: An Introduction to High Dynamic Range Photography*. Short Cut. O'Reilly, March 2007.

[9] Garrett M. Johnson and Mark D. Fairchild. Rendering HDR images. In *IS&T/SID 11th Color Imaging Conference*, pages 36–41, 2003.

[10] Jiangtao Kuang, Hiroshi Yamaguchi, Changmeng Liu, Garrett M. Johnson, and Mark D. Fairchild. Evaluating HDR rendering algorithms. *ACM Transactions on Applied Perception*, 4(2):9, 2007. ISSN 1544-3558.

[11] Patrick Ledda, Greg Ward, and Alan Chalmers. A wide field, hhigh dynamic range, stereographic viewer. In *GRAPHITE '03: Proceedings of the 1st international conference on Computer graphics and interactive techniques in Australasia and South East Asia*, pages 237–244, New York, NY, USA, 2003. ACM. ISBN 1-58113-578-5.

[12] Patrick Ledda, Alan Chalmers, and Helge Seetzen. HDR displays: a validation against reality. In *IEEE International Conference on Systems, Man and Cybernetics*, 2004.

[13] S. Mann. Compositing multiple pictures of the same scene. In *Proceedings of the 46th Annual IS&T Conference*, pages 50–52, Cambridge, Massachusetts, May 9-14 1993. The Society of Imaging Science and Technology. ISBN: 0-89208-171-6.

[14] Ferrell McCollough. *Complete Guide to High Dynamic Range Digital Photography.* Lark Books, May 2008.

[15] Shree K. Nayar and Tomoo Mitsunaga. High dynamic range imaging: Spatially varying pixel exposures. In *Proc. IEEE CVPR*, pages 472–479, 2000.

[16] Erik Reinhard and Paul Debever. *High Dynamic Range Imaging: Acquisition, Display, and Image-based Lighting.* Morgan Kaufmann, 2006.

[17] Erik Reinhard, Michael Stark, Peter Shirley, and James Ferwerda. Photographic tone reproduction for digital images. In *SIGGRAPH '02: Proceedings of the 29th Annual Conference on Computer Graphics and Interactive Techniques*, pages 267–276, New York, NY, USA, 2002. ACM. ISBN 1-58113-521-1.

[18] Allan G. Rempel, Matthew Trentacoste, Helge Seetzen, H. David Young, Wolfgang Heidrich, Lorne Whitehead, and Greg Ward. LDR2HDR: On-the-fly reverse tone mapping of legacy video and photographs. *ACM Transactions on Graphics*, 26(3): 39, 2007.

[19] Helge Seetzen, Lorne A. Whitehead, and Greg Ward. P. 54.2: A high dynamic range display using low and high resolution modulators, May 2003.

[20] Helge Seetzen, Wolfgang Heidrich, Wolfgang Stuerzlinger, Greg Ward, Lorne Whitehead, Matthew Trentacoste, Bhijeet Ghosh, and Andrejs Vorozcovs. High dynamic range display systems. In *SIGGRAPH '04: ACM SIGGRAPH 2004 Papers*, pages 760–768, New York, NY, USA, 2004. ACM.

[21] Jessi Stumpfel, Chris Tchou, Andrew Jones, Tim Hawkins, Andreas Wenger, and Paul Debevec. Direct HDR capture of the sun and sky. In *AFRIGRAPH '04: Proceedings of the 3rd international conference on Computer graphics, virtual reality, visualisation and interaction in Africa*, pages 145–149, New York, NY, USA, 2004. ACM. ISBN 1-58113-863-6.

[22] L. Thurston. A law of comparitive judgment. *Psychology Review*, 34(4):273–286, 1927.

[23] Matthew Trentacoste. Photometric image processing for high dynamic range displays. Master's thesis, The University Of British Columbia, 2006.

[24] Greg Ward. High dynamic range image encodings. [Last Accessed: 1 June 2009], 2005. URL `http://www.anyhere.com/gward/hdrenc/Encodings.pdf`.

[25] Greg Ward. Defining dynamic range. In *SIGGRAPH '08: ACM SIGGRAPH 2008 classes*, pages 1–3, New York, NY, USA, 2008. ACM.

[26] Greg Ward. The hopeful future of high dynamic range imaging: Invited paper. In *SIGGRAPH '08: ACM SIGGRAPH 2008 classes*, pages 1–3, New York, NY, USA, 2008. ACM.

[27] Greg Ward, Maryann Simmons, Brightside Technologies, Walt Disney, and Feature Animation. JPEG-HDR: A backwardscompatible, high dynamic range extension to JPEG. In *Proceedings of the Thirteenth Color Imaging Conference*, pages 283–290, 2005.

[28] Gregory Ward, Holly Rushmeier, and Christine Piatko. A visibility matching tone reproduction operator for high dynamic range scenes. *IEEE Transactions on Visualization and Computer Graphics*, 3:291–306, 1997.

[29] Liz Wells. *Photography: a Critical Introduction.* Routledge, 3 edition, 2004.

[30] Xiaoru Yuan, Minh X. Nguyen, Baoquan Chen, and David H. Porter. HDR VolVis: High dynamic range volume visualization, 2006.