Literature Review: 3D Colour Histogram

Abstract

The preliminary research outlined in this document had the aim of gathering enough information on 3D colour histograms to allow me to begin the planning and development of my project. Five topics of research emerged: colour spaces, colour histograms, prospective technologies, colour quantisation, and colour alterations. I used webpages, academic papers, and existing applications to gather the necessary information to form conclusions that could be applied to my project.

I conclude that either the sRGB and CIE-L*a*b* space are appropriate for my project, and that colour space transformations may be required. From analysis of 3D colour histogram implementations, I have compiled the necessary information to implement my own histogram, and the first stages of the simpler colour quantisation algorithms allow me to plot colours on the histogram. Research into technologies concludes that WebGL, augmented by Three.js, is the most appropriate technology for my project. Finally, analysis of the effects of colour controls concludes that exposure, saturation, and contrast, or lift/gamma/gain will be my colour controls; implementation of these controls can be achieved through the application of lift/gamma/gain or ASC-CDL formulae.

I certify that all material in this dissertation which is not my own work has been identified:

Introduction

The aim of my project is to develop a webpage that allows one to upload an image, view its 3D colour histogram, and make alterations to the image's colour. The research that I have conducted for my project can be organised into five topics: colour spaces, colour histograms, technologies, colour quantisation, and colour alterations. My literature review has been structured to reflect these subject areas. I will now briefly introduce each of these topics, and explain how they relate to my project.

Colour spaces, such as CIE-XYZ and sRGB, are representations of colour within a coordinate system. Each colour space has it's own merits and uses, and when choosing which space to implement, one must consider ones requirements and restrictions. It is important to understand colour spaces because it will allow me to choose an appropriate coordinate system in which to display my colour histograms. Research into this topic also has a second purpose: different applications use different colour spaces, so transforming between them is a process that I am likely to have to implement in my project. Many sources covering colour spaces are webpages with context to digital media, as this is the most common application of colour spaces. Other, more relevant, sources are academic papers that cover the underlying theory of colour spaces and the maths required to transform between them.

Conventional colour histograms plot light levels against their frequency within an image and comprise of three separate plots, one for each colour channel: an example of a colour histogram is shown in §Figure 0:1. Unlike the common 2D histogram, my histogram will be represented in 3 dimensions, by assigning an axis for each colour channel and representing frequency of occurrence by the size of the plot: an example of a 3D colour histogram can be seen in §Figure 0:2. 2D colour histograms do not apply directly to my project, but they do provide some background information on colour histograms; sources also mostly refer to digital photography, and are therefore of little relevance to my project. While there is little writing about 3D colour histograms, implementations of 3D colour histograms, which can be analysed and learnt from, are of relevance to my application.

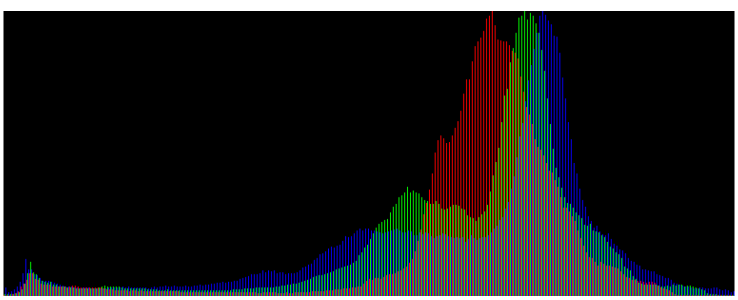

Figure 0:1 Colour Histogram
(https://upload.wikimedia.org/wikipedia/commons/f/f5/Odd-eyed_cat_histogram.png)

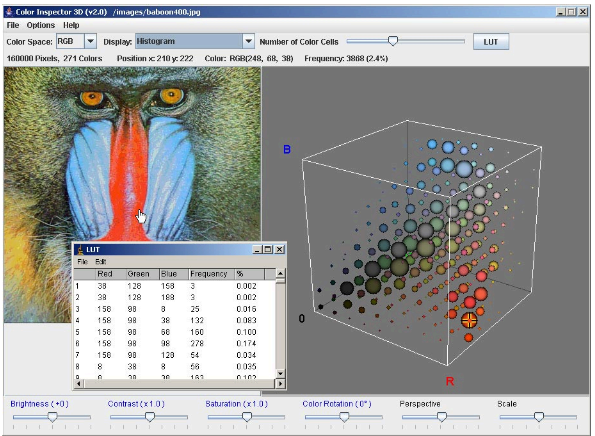

Figure 0:2 3D Colour Histogram (http://rsb.info.nih.gov/ij/plugins/images/3d-inspector.jpg)

Another consideration that must be made is which technologies will be used in project. My application will be web-based, so any technology that can be incorporated into HTML and JavaScript is ideal. A lot of the code for my application will be focused on the renderings, so most of my research will be into 3D rendering technologies. Most of the sources take the form of online tutorials and library documentations, as these give a good impression of a technology's capabilities and usability. I will apply my findings directly to my project by using my chosen technologies in the development of my application.

An essential process in my project is reading colours from an image into a format that can be displayed by a histogram: this can be achieved by a process called colour quantisation. Colour quantisation is applied to compress images by reducing the number of colours in their palette while loosing the least amount of visual information as possible. Full colour quantisation is not relevant to my project, as it is not necessary to redraw an image with a smaller palette, or prioritise colours; instead I will use the early stages of colour quantisation algorithms to distribute the colours in an image into an array of bins that can be displayed as plots on a histogram. Most resources on colour quantisation are irrelevant, as they are in context of other applications of colour quantisation, but useful information can still be extracted and some sources describe similar algorithms, which are applicable to my project.

The final area of research for my project is into colour adjustments and how to implement them. I will research colour alteration tools by using existing image editing software and analysing the effects of controls; I will then research how to implement these controls. Sources on colour alterations are mostly either existing image editing software or technical descriptions of colour alteration functions. Findings from this research will be applied to my project by helping me to decide what controls I will include in my application and how I will implement them.

1. Colour Spaces

Colour spaces are a way in which colours can be defined within a coordinate system; they allow colours to be referred to and transformed within or between colour spaces. The research into colour spaces for my project falls into two categories: which colour space or spaces to implement, and how to transform between them. Sources for the former category are mostly webpages and academic papers on colour science; sources for the latter are more applied, so consist of instructional webpages, often containing pseudo-code and formulae.

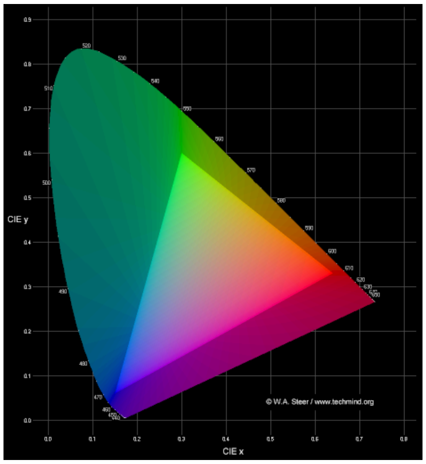

Figure 1:1 1931 CIE Chromaticity Diagram (http://www.techmind.org/colour/ciexymedium.png)

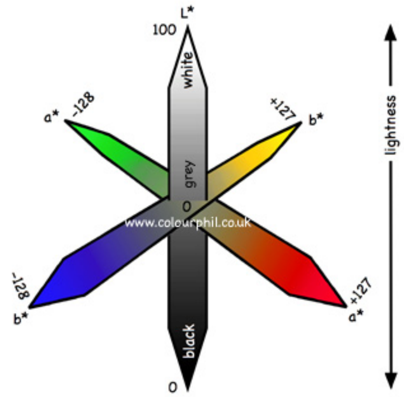

Figure 1:2 CIE-L*a*b* (http://www.colourphil.co.uk/images/lab_colourspace_3.jpg)

1.1. CIE 1931 XYZ Colour Space

In the late 1920's, William David Wright conducted experiments into the human eye's sensitivity to colour; from these experiments the CIE-XYZ colour space emerged (Wright, 1929). Andrew Steer's introduction to colour science (Steer, 2008) gives a clear and concise explanation of the CIE-XYZ colour space, as well as how it relates to other colour spaces and gamuts; Professor Gernot Hoffmann, on the other hand, goes into a lot more detail in her paper on the subject (Hoffmann, CIE Color Space, 2015).

Much of the information in both Steer and Hoffmann's writings is useful in the understanding of how CIE-XYZ was created and how it represents colours, but what is of particular interest is how CIE-XYZ relates to other colour spaces and gamuts. §Figure 1:1 from Steer's webpage, depicts how the sRGB gamut fits within the CIE-XYZ colour space. As Steer explains, the CIE-XYZ colour space is a three dimensional space in which all perceivable colours can be represented; the sRGB gamut, which is the range of colours displayed by most monitors, can therefore fit within the CIE-XYZ space. This is of significance because if all other colour spaces and gamuts can fit within CIE-XYZ, they can be transformed in and out of it, making it the common colour space for many transformations between spaces.

1.2. CIE-L*a*b*

While researching colour spaces and colour perception, I discovered that the perceptual difference between colours in the sRGB space is not linear: that is to say that the change in colour perceived over an arbitrary distance will vary in different locations throughout the colour space. This can cause colour representations to look inaccurate and reduces the effectiveness of the representation. An alternative representation to RGB is CIE-L*a*b*, which has a more uniform perceived colour change, making it a good candidate for the colour space in which to display a histogram.

Phil Cruse's webpage on the CIE-L*a*b* colour space (Cruse, 2015) describes the anatomy of the space, depicted in §Figure 1:2, as well as its applications. His webpage serves as a good introduction to the colour space and I conclude that it is a good candidate colour space to represent my histogram in, but contains little technical detail that would help me to implement it.

Professor Hoffmann also wrote a paper on the subject of the CIE-L*a*b* colour space (Hoffmann, CIELab Color Space, 2015), and like her paper on the CIE-XYZ colour space, it gives a concise introduction to the colour space as well as technical details. Chapters of Hoffmann's paper that are of particular interest are those concerning transformations in and out of CIE-L*a*b*, which may be applied in my project if I were to implement colour space transformations.

1.3. Colour Space Transformations

Because any colour space can be transformed into CIE-XYZ, it is often used as a common space when transforming between two other spaces or gamuts. If I decide to represent my histogram in a colour space other than sRGB, which input colours are likely to be in, I will need to perform a transformation and therefore use CIE-XYZ.

Ryan Juckett's webpage, RGB Color Space Conversion (Juckett, 2010), describes how to define an RGB colour gamut using CIE-XYZ and includes useful information, such primary coordinates, white points, gamma correction curves, and transformation matrices. Juckett also provides GLSL code for the transformation between sRGB and CIE-XYZ, which is likely to be useful in my project because GLSL is the language used by WebGL to run scripts on the local GPU. Professor Hoffmann's two papers (Hoffmann, CIE Color Space, 2015) (Hoffmann, CIELab Color Space, 2015) also include descriptions of the transformations between colour spaces. From Juckett and Hoffmann's work, I conclude that CIE-XYZ will be integral to my project, should I be required to perform colour space transformations.

Juckett and Hoffmann only provide formulae of transformations between the sRGB, CIE-XYZ, and CIE-L*a*b* sapces; Irotek's website on colour space transformation (Irotek Group Ltd., 2014) provides pseudo-code formulae for transformations between many 3D colour spaces and gamuts and these will be valuable for reference if I am to use other colour spaces, such as HSV/HSL.

2. Colour Histograms

The next avenue of research for my project is into colour histograms. The webpage that inspired my project was Javier Villarroal's 3D colour histogram (Villarroal, 2013), but before I discuss his work I will cover some background information on colour histograms. Luminosity and colour histograms are widely used in digital photography, so due to their applied nature, many of my sources are aimed at photographers and have limited relevance to my project. Other sources, such as implementations of 3D histograms, are relevant to my project and can be analysed to extract useful conclusions.

2.1. 2D Colour Histograms

The Practical Photography Tips website has an informative page on colour histograms (PracticalPhotographyTips) that covers what they are and how they are used in photography: however this source has limited value because of its target audience and applied nature. Ken Rockwell's webpage on colour histogram applications in photography (Rockwell, 2006) and the Cambridge in Colour webpage on colour and luminance histograms (McHugh, 2015) are two other sources that introduce the concept of histograms, but contain a limited amount of relevant information.

2.2. 3D Colour Histograms

My project will implement a 3D colour histogram, which unlike its 2D counterparts assigns a colour channel to each axis. Javier Villarroal's webpage (Villarroal, 2013) is a good example of a 3D histogram and was the inspiration for my project. This resource is integral to my project as it is what my prototype is based on and allows me to have a clear vision of where my project is going. Since finding Javier's website I have done further research into 3D colour histograms and found that there is little writing on the subject, but instead there are several implementations to be found online: Kai Uwe Barthel's histogram, implemented in Java (Barthel, 2007); Third Avenue Design's web based histogram (Third Ave Design); and a real-time video histogram (Caballero & Belmonte). I will conduct research in this area by analysing each implementation and use my conclusions to form the specifications for my own histogram.

From my analysis of Javier's histogram, I conclude that it has 2 advantages over others: it is web-based, and therefore more accessible; and its design is intuitive and can be customised by the user. Javier's histogram has limitations: it does not allow one to upload ones own images and instead offers a selection of stock photos. Third Avenue Design's histogram is web-based and intuitive to use, but it also allows one to upload ones own images: however it does not provide allow the user to edit their images, unlike Barthel's Java implementation. Each histogram so far been implemented in the RGB colour space, but Caballero and Belmonte's histogram, capable of displaying a histogram of video in real-time, allows one to choose the colour space in which the histogram is viewed; I intend for my histogram to only be implemented in one colour space, at least initially, but Caballero's webpage allows me to compare colour spaces and make a more informed decision. By analysing these four histograms I can conclude that my histogram will be web-based, with an intuitive design similar to Javier's, will allow the user to upload their own images as well as apply colour alterations to that image; Caballero and Belmonte's histogram will allow me to compare colour spaces and choose the one most appropriate for my project.

3. Technologies

I chose to develop my project as a webpage to increase its accessibility and convenience: this does, however, limit my choice of technologies to ones that can be incorporated into HTML and JavaScript. I have previous experience in creating web-based, 3D renderings in WebGL, a JavaScript adaptation of the popular 3D rendering platform, OpenGL. My research into technologies consists of online tutorials and API documentations, as these allow me to form my own opinions on which will be most appropriate.

3.1. WebGL & Three.js

The technology that I chose to develop the bulk of my project with is WebGL, a "cross-platform, royalty-free web standard for a low-level 3D graphics API based on OpenGL" (Khronos Group). WebGL utilises the HTML5 Canvas element, is written using JavaScript, and is widely used: from these accolades I concluded that I would use WebGL as my primary technology.

The WebGL wiki (Khoronos Group, 2015) provides an introduction into WebGL as well as tutorials and demonstrations; additional tutorials can be found on Tony Parisi's website, Learning WebGL (Parisi, 2010). After working through these tutorials I came to the conclusion that WebGL code is bloated and using it to develop a complex project could be problematic. I conducted further research and found a JavaScript library called Three.js (mrdoob, 2015) that wraps up WebGL's functionality into a concise and easily understood framework. Documentation and examples for Three.js can be found on their website (mrdoob, three.js, 2015) and demonstrate the relative simplicity of Three.js compared to WebGL, leading me to the conclusion that I shall also use Three.js in the development of my project.

3.2. Fragment Shaders and GLSL

I have previous experience with WebGL and Three.js so know some of the obstacles that will need to be overcome during my project; one such obstacle is how to independently colour objects in a rendering if they are contained with the same geometry. One solution is to write a small program in GLSL, OpenGL's shading language, which is similar to C/C++ and designed for matrix and vector manipulation.

Javi Fontan's website (Javi Fontan, 2015) allows one to write GLSL programs in the browser and view their output as the webpage's background. It also contains some prewritten examples, demonstrating GLSL's capabilities. Debugging in GLSL is difficult as one can't write to a console, so Fontan's website is valuable when writing small programs before incorporating them into the main program.

To run a GLSL program in Three.js, one must use Three.js' Shader Material, which fetches the GLSL code and applies it as a material to a geometry. The Three.js documentation for the Shader Material (mrdoob, ShaderMaterial) explains how to use it, and in particular how to pass arguments into the GLSL program, such as an array of points. I will be referring to this documentation when it comes to the point that I have to implement a Shader Material.

4. Colour Quantisation

In a given image it is unlikely that there will be many pixels with exactly the same colour; to prevent the histogram from being too widely dispersed it is necessary to distribute pixels into ranges of colours, or bins, that can be plotted on the histogram as a single point. Colour quantisation is the process of reducing the size of an image's palette for the sake of compression and can be utilised for my application.

4.1. Colour Quantisation for Compression

In his summary of colour quantisation techniques, Segenchuk (Segenchuk, 1997) discusses four approaches to colour quantisation, two of which, median-cut and octree, are explored in more depth in Dan Bloomberg's webpage on colour quantisation (Bloomberg, 2008). From Segenchuk's overview and Bloomberg's detailed descriptions, one can conclude that the most successful colour quantisation algorithms do not divide the colour space into uniform segments, but rather prioritise those with the highest concentration of pixels, so that the smaller palette of colours is used as effectively as possible. These advanced techniques produce good results when recreating an image with a smaller colour palette but my project requires evenly divided bins, so these techniques are not applicable.

4.2. Colour Quantisation for Histograms

Khouloud Meskaldji et. al.'s paper on the effects of colour quantisation on colour histogram based image searches (Meskaldji, Boucherkha, & Chikhi, 2009) explains the process of colour quantisation and how it can be applied to colour histograms. Initially this source looks promising but because their work is focused on image searching, most of the paper is irrelevant to my project and there is no technical detail on implementing colour quantisation.

In his overview of colour quantisation techniques, Steven Segenchuk (Segenchuk, 1997) describes four techniques; as covered in §4.1, most of these techniques aren't suited to my application, but one them, uniform quantisation, is applicable to my project. Uniform quantisation divides each axis into uniform segments and assigns pixels to their closest segment. Further research into uniform quantisation yields Eric Liao's uniform colour quantisation pseudo code (Liao, 2013), from which I conclude that it is simple to implement and relevant to my project. Liao's pseudo code and accompanying documentation is informative enough to allow me to implement it in my project.

5. Colour Alterations

My project will allow the user to apply adjustments to the colours of their image and observe the effects on the histogram in real-time. My research into colour alterations comprises of two phases: analysing several colour controls and identifying those that are most applicable to my project, and researching how to implement them. I have used image-editing software to identify colour controls, and technical websites to research their implementations.

5.1. Analysis of Controls

Apple's iPhoto application on OSX (Apple Inc., 2015) allows one to edit images, including making colour alternations. The available colour controls in iPhoto are grouped into three categories, the contents of which are: exposure, contrast, and saturation; definition, highlights, shadows, sharpness, and de-noise; and temperature and tint. These controls are accompanied by a 2D colour histogram, which allows me to analyse of effects of each control and therefore decide which will be the most applicable to my project.

Controls in the first category, namely exposure, contrast, and saturation, have a considerable effect on the histogram: exposure compresses or stretches all channels to the left or right; contrast stretches and compresses channels around the x-axis mid-point; and saturation separates or converges channels. From these observations, I conclude that exposure, contrast, and saturation are good candidates for colour controls in my application.

Conversely, the sharpness and de-noise controls have very little effect on the histogram and are not appropriate for my project. The highlights and shadows controls have a noticeable effect on the histogram, shifting it left and right respectively. These effects are similar to those seen in software that I have used previous and I will consider them for use in my application.

The final two colour controls available in iPhoto are tint and temperature; these controls, unlike previous ones, effect hue: tint moves colours towards the magenta and green hues, while temperature moves them towards the blue or yellow hues. These changes have significant effect on the histogram and would be appropriate for my project: however they are not as commonly used as other controls, so they may not take priority over, more useful controls.

5.2. ASC-CDL and Lift/Gamma/Gain

The highlights and shadows controls in iPhoto have a similar effect on the histogram to controls that I have used previously in Sony RAW Viewer (Sony Corporation, 2015), known as lift/gamma/gain, that correspond to highlights, mid-tones, and shadows respectively. These three components are applied to input colours using a formula and alter the colours within an image.

Lift/gamma/gain is one of the colour correction schemes discussed by Alexis Han Hurkman on his webpage (Hurkman, 2013); Hurkman discusses its applications and how it is a versatile tool for altering the colours of an image. One of Hurkman's diagrams, seen in §Figure 5:1, shows how each component effects different tones: the x-axis is a tonal range from black to white, and the y-axis is the magnitude of that component's effect. On one of the Blender wiki's webpages, written by the user Xat (Xat, 2010), one can find the formula used to apply lift/gamma/gain, and it is displayed in §Figure 5:2; this formula could be used in my project to apply the lift/gamma/gain adjustments to an image.

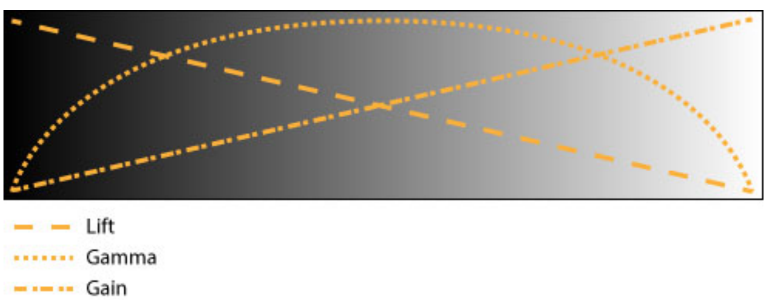

Figure 5:1 Effect of lift/gamma/gain (http://content.provideocoalition.com/uploads/AVH/Lift-Gamma-Gain.jpg)

$$out = (gain * (in + lift * (1 - input)) * (1 / gamma)$$

Figure 5:2 Lift/gamma/gain Formula

In his webpage, Hurkman discusses another colour adjustment scheme called ASC-CDL; ASC-CDL (Colour Decision List) applies the three parameters, slope, offset, and power, to colours within an image. Xat's webpage on the Blender Wiki also has the ASC-CDL formula, which can be seen in §Figure 5:6. Nikolali's webpage introducing CDL (Nikolai, 2013) provides a series of graphs, seen in §Figure 5:3, §Figure 5:4, and §Figure 5:5, which, like Hurkman's graph in §Figure 5:1, show the effects of each component at different tones. ASC-CDL is another method of colour alteration that could be applied to my project instead of lift/gamma/gain.

Francois Tarlier's comparison of lift/gamma/gain and CDL (Tarlier, 2009) is an applied viewpoint of the two: his conclusion, that CDL is easier to use but lift/gamma/gain allows more control, is useful to consider from the point of view of what users will want, but for my project it is still unclear which would be more applicable.

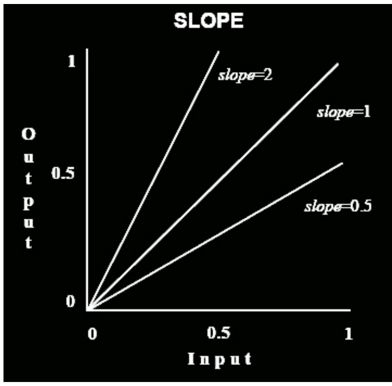

Figure 5:3 Slope (http://www.niwa.nu/wp-content/uploads/2013/05/Slope.jpg)

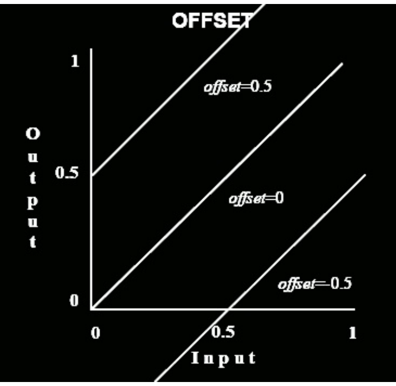

Figure 5:4 Offset (http://www.niwa.nu/wp-content/uploads/2013/05/Offset.jpg)

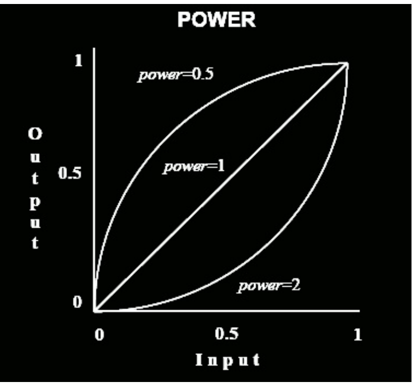

Figure 5:5 Power (http://www.niwa.nu/wp-content/uploads/2013/05/Power.jpg)

*output = (input * slope + offset)* ^{power}

Figure 5:3 ASC-CDL Formula (http://wiki.blender.org/index.php/User:Xat/KeyerNode)

Conclusion

In my research into colour spaces and transformations, sources fell into two categories: those that are useful in the comprehension of the subject and in choosing a colour space to represent histograms in, and those containing applicable information on colour space transformations. Andrew Steer's introduction into colour science (Steer, 2008) and Phil Cruse's overview of the CIE-L*a*b* colour space (Cruse, 2015) both allowed me conclude that sRGB and CIE-L*a*b* are two candidate colour spaces in which to represent my histograms. Other resources, such as Ryan Juckett's webpage (Juckett, 2010), Professor Hoffmann's papers on the CIE-XYZ and CIE-L*a*b* colour spaces (Hoffmann, CIE Color Space, 2015) (Hoffmann, CIELab Color Space, 2015), and Irotek's list of conversion formulae (Irotek Group Ltd., 2014) contain details on colour space transformations, and because I am likely to perform transformations in my project, these resources will be applicable to my application.

Most applications of 2D colour histograms are in digital photography, while sources relating to 3D histograms are mostly implementations. Practical Photography Tips (PracticalPhotographyTips), Rockwell's webpage (Rockwell, 2006), and the Cambridge in Colour webpage (McHugh, 2015) give a good introduction to colour histograms, but due to their applied nature they have little application in my project. After analysing implementations of 3D histograms, such as Villarroal's (Villarroal, 2013), Third Avenue Design's histogram (Third Ave Design), Caballero's real-time video histogram (Caballero & Belmonte), and Barthel's Java program (Barthel, 2007) I was able to come to conclusions about 3D colour histograms: in particular which features will be applied in my project.

Analysing tutorials and documentations for WebGL and Three.js brought me to the conclusion that using Three.js would be easier and quicker, and further research lead me to the conclusion that I would also have to use GLSL shaders during my project. Parisi's and the Khoronos Group tutorials (Parisi, 2010) (Khoronos Group, 2015) serve to demonstrate my reasons for using Three.js, but will not be used directly in my project. Other resources, such as the Three.js documentation (mrdoob, three.js, 2015) (mrdoob, ShaderMaterial) and the in-browser GLSL compiler (Javi Fontan, 2015) will be used for reference in the development of my project.

Preliminary research found that colour quantisation could be used to categorise colours within an image. No useful conclusions were made from analysing Khouloud Meskaldji et. al.'s paper on the effects of colour quantisation on colour histograms. Steven Segenchuk's webpage (Segenchuk, 1997) gives explanations of a variety of colour quantisation techniques, but I concluded that only the simplest are applicable to my project; Eric Liao provides a pseudo-code implementation of one of these algorithms on GitHub (Liao, 2013) and it is likely that Liao's pseudo-code will be applied during the development of my colour quantisation algorithm.

While I have drawn a number of conclusions from my research into colour alterations, I am still yet to decide which controls and implementations I will use. From experimenting in Apple's iPhoto application (Apple Inc., 2015) I have identified exposure, contrast, and saturation, and lift/gamma/gain as candidate controls for my project; from previous experience in colour manipulation I identified ASC-CDL and lift/gamma/gain as potential implementations of these controls. From Hurkman's summary of lift/gamma/gain (Hurkman, 2013) and Nikolai's webpage on CDL (Nikolai, 2013) I know how to apply each scheme. I am yet to draw any conclusions from Tarlier's comparison of the ASC-CDL and lift/gamma/gain (Tarlier, 2009), as its comparison of the two schemes is superficial. While I still have decisions to make regarding colour alterations, sources compiled and conclusions made will allow me to choose appropriate controls and implement them in my project.

Bibliography

Apple Inc. (2015, March 19). iPhoto. *9.6.1*. Apple Inc.

Barthel, K. U. (2007, September 16). *3D Color Inspecter/Color Histogram.* Retrieved November 12, 2015 from Research Services Branch: http://rsb.info.nih.gov/ij/plugins/color-inspector.html

Bloomberg, D. (2008, September 4). *Color Quantization*. Retrieved November 10, 2015 from Leptonica: http://www.leptonica.com/color-quantization.html

Caballero, M. L., & Belmonte, N. G. (n.d.). *Real time color 3D histogram analysis*. Retrieved October 3, 2015 from Senchalabs: http://www.senchalabs.org/philogl/PhiloGL/examples/histogram/

Cruse, P. (2015, June 12). *Introduction to Colour Spaces.* Retrieved October 2, 2015 from Colour Phil: http://www.colourphil.co.uk/lab_lch_colour_space.shtml

Hoffmann, G. (2015, September 24). *CIE Color Space.* Retrieved November 5, 2015 from docs-hoffman: http://docs-hoffmann.de/ciexyz29082000.pdf

Hoffmann, G. (2015, September 24). *CIELab Color Space.* Retrieved November 5, 2015 from docs-hoffmann: http://docs-hoffmann.de/cielab03022003.pdf

Hurkman, A. H. (2013, May 20). *What's In a Name?* Retrieved November 9, 2015 from Pro Video Coalition: http://www.provideocoalition.com/whats-in-a-name

Irotek Group Ltd. (2014). *Color conversion math and formulas*. Retrieved October 2, 2015 from EasyRGB: http://www.easyrgb.com/index.php?X=MATH&H=07#text7

Javi Fontan, D. G. (2015, April 10). *GLSL Sandbox*. Retrieved October 2, 2015 from GLSL Sandbox: http://glslsandbox.com/

Juckett, R. (2010, May 16). *RGB Color Space Conversion*. From Ryan Juckett: http://www.ryanjuckett.com/programming/rgb-color-space-conversion/

Khoronos Group. (2015, April 26). *Main Page*. Retrieved November 5, 2015 from WebGL Wiki: https://www.khronos.org/webgl/wiki/Main_Page

Khronos Group. (n.d.). *WebGL*. Retrieved November 5, 2015 from Khronos: https://www.khronos.org/webgl/

Liao, E. (2013, April 18). *Uniform Color Quantization*. Retrieved November 8, 2015 from Github: https://github.com/rcliao/MultimediaSystem/wiki/Uniform-Color-Quantization

McHugh, S. (2015). *Camera Histograms: Luminosity & Color*. Retrieved October 3, 2015 from Cambridge in Colour: http://www.cambridgeincolour.com/tutorials/histograms2.htm

Meskaldji, K., Boucherkha, S., & Chikhi, S. (2009). Color Quantization and its Impact on Color Histogram Based Image Retrieval. In O. A. Abdelmalek Amine (Ed.), *Conférence Internationale sur l'informatique et ses Applications. 547.* Algeria: http://ceur-ws.org.

mrdoob. (n.d.). *ShaderMaterial*. Retrieved October 2, 2015 from three.js: http://threejs.org/docs/#Reference/Materials/ShaderMaterial

mrdoob. (2015, September 22). *three.js*. Retrieved September 27, 2015 from three.js: http://threejs.org/

Nikolai. (2013, May 9). *Color Decision List - introduction*. Retrieved October 2, 2015 from Waldman Media: http://www.niwa.nu/2013/05/color-decision-list-introduction/

Parisi, T. (2010, December 6). *The Lessons*. (T. Parisi, Editor) Retrieved October 2, 2015 from Learning WebGL: http://learningwebgl.com/blog/?page_id=1217

PracticalPhotographyTips. (n.d.). *Color Histograms*. Retrieved October 31, 2015 from Practical Photography Tips: http://practicalphotographytips.com/Digital-Camera-Basics/color-histograms.html

Rockwell, K. (2006). *How To Use Color Histograms*. Retrieved October 3, 2015 from Ken Rockwell: http://www.kenrockwell.com/tech/yrgb.htm

Segenchuk, S. (1997, May 5). *An Overview of Colour Quantization Techniques*. Retrieved November 7, 2015 from Worcester Polytechnic Institute: http://web.cs.wpi.edu/~matt/courses/cs563/talks/color_quant/CQindex.html

Sony Corporation. (2015, March 9). RAW Viewer. *2.2.2*. Sony Corporation.

Steer, W. A. (2008, March 15). *Introduction to Colour Science*. Retrieved November 5, 2015 from Techmind: http://www.techmind.org/colour/

Tarlier, F. (2009, November 5). *lift/gamma/gain VS. ASC CDL*. Retrieved October 2, 2015 from Francois Tarlier's blog: http://www.francois-tarlier.com/blog/liftgammagain-vs-asc-cdl/

Third Ave Design, T. S. (n.d.). *3D Histogram*. Retrieved October 3, 2015 from 3D Histogram: http://3dhistogram.com/?#contact

Villarroal, J. (2013). *H3Stogram - 3D Interactive Color Histogram*. Retrieved September 27, 2015 from H3Stogram: http://h3stogram.herokuapp.com/

Wright, W. D. (1929, February 7). A re-determination of the trichromatic coefficients of the spectral colours. *Transactions of the Optical Society , 30* (4), pp. 141-164.

Xat. (2010, Janurary 15). *Sequencer color correction*. Retrieved November 11, 2015 from Blender Wiki: http://wiki.blender.org/index.php/User:Xat/KeyerNode