Integrating Colour Correction Algorithms

Fufu Fang

School of Computing Sciences
University of East Anglia

This dissertation is submitted for the degree of
Doctor of Philosophy

December 2021

This copy of the thesis has been supplied on condition that anyone who consults it is understood to recognise that its copyright rests with the author and that use of any information derived there from must be in accordance with current UK Copyright Law. In addition, any quotation or extract must include full attribution.

Acknowledgements

First and foremost, I wish to thank my primary supervisor Graham Finlayson, for giving me this opportunity to study Colour Sciences, for his generous and exceptional support. Graham has been an excellent supervisor as well as a good friend. I have always found our trips to scenic location such as Eagle Park inspiring.

I would also like to thank everyone who have been on my supervisory team – Michal Mackiewicz, Javier Vazquez-Corral, Han Gong and Mark Fisher. I struggled a lot with thesis writing. I want to especially thank Graham, Michal and Javier – thank you for the exceptional help you provided. Thank you for pushing me. Without your support the thesis probably would not have been completed.

I would also like to thank Neal Carr for providing invaluable mathematical advice and being a loyal friend. Thank you for making multidimensional linear algebra a bit easier to understand. Thank you for always answering my phone call. Talking to you is always fun.

My education has been a long and arduous journey – originally I did not even set out to study computer science. I would like to take this opportunity to acknowledge those who have made a significant impact in my life.

I studied my undergraduate Computer Science degree in University of York. I would like to thank my undergraduate final year project supervisor Will Smith. Thank you for answering that email all those years ago while you were the Admission Tutor, and giving me a place during Clearing. Hopefully I have made you proud.

I would like to thank Julian Willis, for being such a wise, kind and fun friend over all these years. I have learnt a lot from you. Thank you for always being so calm and reassuring. Thank you for providing so much practical and useful advice. You are so good at hosting parties. I have always enjoyed my semi-annual visit to your home in Cambridge. Being able to help out during your PhD and eventually making a guest appearance in your thesis is definitely a highlight of my time as a PhD student. Thank you for providing me with the opportunity.

I would also like to thank Bethany Plummer, for being such a caring and lovely friend. Thank you for comforting me when I am upset. Thank you for giving me video calls over the lockdown. Thank you for arranging so many fun and incredible activities. I loved of all those

birthday celebrations you organised, especially our trip to Wales. I really enjoyed all the cycling trips we did together. I can't believe we have done two 100-mile bike rides together!

I would also like to thank Alan Sydee and Sarah Palmer. Alan, thank you for re-introducing me to cycling when I first arrived in Norwich and helping out with all my house moves during my time in Norwich. Sarah, thank you for inviting me to celebrate Christmas with your family. Hopefully when you two have children, they will like Settler of Catan as much as we do.

I would also like to thank Callum Rollo and Guilly Rident. Thank you for always welcoming me to your home and being my official lockdown bubble. Thank you for all the delicious vegetarian meal and thought-provoking conversations. Callum, I want to especially thank you for teaching me how to kayak properly, I have really enjoyed our time on River Wensum.

I would also like to thank my old housemates Rachael Watts and Matthew Kao. Thank you for always being so welcoming, fun and generous. Thanks for always being there when I need someone to talk to. Rachael, I wish you luck in your own PhD journey!

I would like to thank every member of the UEA Colour Group past or present who have helped me and given me good memories. In particular, I would like to thank Ghalia Hemrit for being a good friend. Thank you for giving me encouragement and commenting on every single one of my Instagram story. I would like to thank Alex Hayes for all the good advices and fun conversations. I really do love visiting you and sampling all the cuisines in those interesting restaurants. Ellie, thank for you allowing me to practice my wild survival skills in your garden. During that short stay in your family home, I really felt I was Bear Grylls. Yuteng, thank you for always providing good contrasting opinions and bringing me down to earth when my ideas run wild.

Finally, and most importantly, I would like to thank my parents for all the love and support.

Abstract

Digital cameras sense colour different than the human visual system (HVS). Digital cameras sense colour using imaging sensor, whereas the HVS senses colour using the cone photoreceptors in our retina. Each digital camera model has its own device specific spectral sensitivity function. It is therefore necessary to convert the device specific colour responses of an imaging sensor to values that are related to the HVS. This process is typically referred to as *colour correction*, and it is common to the image processing pipeline across all cameras.

In this thesis, we explore the topic of colour correction for digital cameras. Colour correction algorithms establish the mapping between device specific responses of the camera with HVS related colour responses. Colour correction algorithms typically need to be trained with datasets. During the training process, we adjust the parameters of the colour correction algorithm, in order to minimise the fitting error between the device specific responses and the corresponding HVS responses.

In this thesis, we first show that the choice of the training dataset affects the performance of the colour correction algorithm. Then, we propose to circumvent this problem by considering a reflectance dataset as a set of samples of a much larger reflectance space. We approximate the convex closure of the reflectance dataset in the reflectance space using a hypercube. Finally we integrate over this hypercube in order to calculate a matrix for linear colour correction. By computing the linear colour correction matrix this way, we are able to fill in the gap within a reflectance dataset.

We then expand upon the idea of reflectance space further, by allowing all possible reflectances. We explore an alternative formulation of Maximum Ignorance with Positivity (MIP) colour correction. Our alternative formulation allows us to develop a polynomial variant of the concept. Polynomial MIP colour correction is far more complex thant MIP colour correction in terms of formulation. Our contribution is theoretically interesting, however practically, it delivers poorer performance.

Access Condition and Agreement

Each deposit in UEA Digital Repository is protected by copyright and other intellectual property rights, and duplication or sale of all or part of any of the Data Collections is not permitted, except that material may be duplicated by you for your research use or for educational purposes in electronic or print form. You must obtain permission from the copyright holder, usually the author, for any other use. Exceptions only apply where a deposit may be explicitly provided under a stated licence, such as a Creative Commons licence or Open Government licence.

Electronic or print copies may not be offered, whether for sale or otherwise to anyone, unless explicitly stated under a Creative Commons or Open Government license. Unauthorised reproduction, editing or reformatting for resale purposes is explicitly prohibited (except where approved by the copyright holder themselves) and UEA reserves the right to take immediate ‘take down’ action on behalf of the copyright and/or rights holder if this Access condition of the UEA Digital Repository is breached. Any material in this database has been supplied on the understanding that it is copyright material and that no quotation from the material may be published without proper acknowledgement.

Publications

The following publications are related to this thesis:

- Fang, F, Gong, H, Mackiewicz, M and Finlayson, G. D. (2017) Colour Correction Toolbox, *Proceedings of 13th AIC Congress 2017.* Korea Society of Color Studies, Jeju, Korea.
- Fang, F and Finlayson, G. D. (2017) Maximum Ignorance Polynomial Colour Correction, *Proceedings of 13th AIC Congress 2017.* Korea Society of Color Studies, Jeju, Korea.
- Finlayson, G. D., Vazquez-Corral, J., Fang, F. (2021). The Discrete Cosine Maximum Ignorance Assumption. *Color and Imaging Conference 2021* Society for Imaging Science and Technology, 2021

Table of contents

List of figures

List of tables

Chapter 1

Introduction

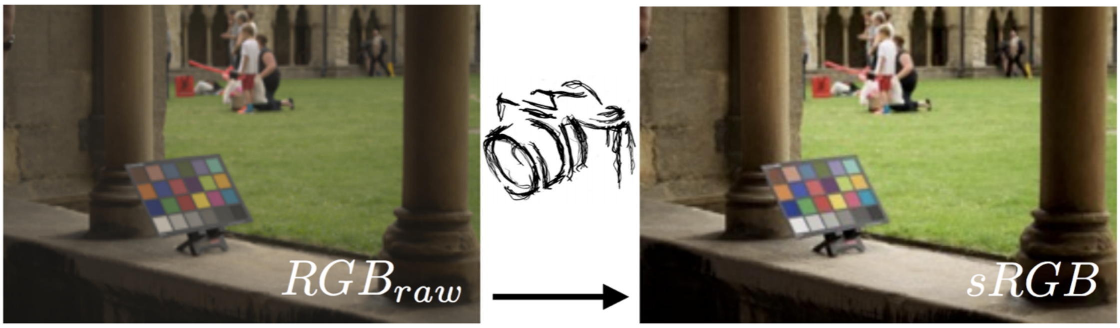
Digital cameras are becoming more common as they become increasingly affordable. The price of a new compact digital camera can be as low as £70. The price of a new entry-level Digital Single-Lens Reflex (DSLR) camera is at around £300. Perhaps the most widely used digital cameras are the built-in cameras of smartphones. Smartphones are one of the must-have accessories in the modern society. Most smartphones typically come with a digital camera. Some of the high-end smartphones even come with multiple cameras different features, e.g. wide-angle lens or zoom lens.

Digital camera-related research topics have become important, due to their widespread use. Colour is a major research topic, as the overall quality of an image is affected the quality of its colour. Capturing images that are pleasing to the human visual system (HVS) is not a straightforward topic, as digital cameras sense colour differently than HVS. Digital cameras sense colour using their imaging sensors, whereas HVS sense colour using the retina. This is because the imaging sensors and the cones (the light sensitive cells in the retina) have different spectral sensitivity functions. This difference is due the limitation and difficulties in imaging sensor design and fabrication. Furthermore, different imaging sensors have different sensitivity functions, as different imaging sensors have different design priorities.

For capturing images, it is necessary to convert the colour responses from the camera into a form that is related to the colour responses of HVS. This process, whicch is common to the image processing pipelines of across all digital cameras, is called colour correction.

In Figure 1.1, we show a typical colour correction scenario. We take a picture of the Macbeth ColorChecker and then regress the camera space RGBs to the display space RGBs, so the captured image looks as close as possible to the correct colours, i.e. the colours we see ourselves when viewing the checker.

In this thesis, we explore the topic of colour correction for digital cameras. There are many colour correction algorithms. They had been designed to solve different optimisations.



$$\begin{bmatrix} R_1 & G_1 & B_1 \\ R_2 & G_2 & B_2 \\ \vdots & \vdots & \vdots \\ R_{24} & G_{24} & B_{24} \end{bmatrix} \begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{bmatrix} \approx \begin{bmatrix} r_1 & g_1 & b_1 \\ r_2 & g_2 & b_2 \\ \vdots & \vdots & \vdots \\ r_{24} & g_{24} & b_{24} \end{bmatrix}$$

Fig. 1.1 The top-left and top-right shows respectively raw and colour corrected image. SRGB gamma have been applied to both images). Prior to applying the gamma correction, the raw values were mapped into RGB values using a linear 3 × 3 matrix. This matrix is chosen to best fit the data.

The difference in algorithms is influenced by various trade-offs, e.g. insisting that white is exactly corrected at the cost of a higher colour error overall, ensuring that a correction transform works for different exposures and etc. Typically a colour correction algorithm is trained before a digital camera leaves the factory. We say “trained” because a typical colour correction algorithm is data-driven optimisation. The data which feed the optimisation depend on the spectral sensitivities of the camera in question. Not only cameras of different models have different spectral sensitivities, cameras of the same model will have slightly different sensitivities if they were manufactured in a different batch.

The training process for a typical colour correction algorithm is carried out using training targets. The quality of training target affects the performance of the algorithm. During training, we map camera measurements to corresponding HVS coordinates while minimising some fitting error by adjust the parameters of the algorithm. The trained algorithm is guaranteed to be optimal for the training dataset but may be sub-optimal for other reflectance samples that the camera may encounter in real-world usage.

In this thesis, we explore the topic of colour correction algorithms. We evaluate both the performance of the colour correction algorithms as well as the reflectance training datasets. In particular, we attempt to address two questions in this thesis – “what reflectance dataset to use” and “how many samples are required”. These two questions are important, because when comparing colour correction algorithms, their performances are affected by the choices of the reflectance datasets. This thesis is therefore useful for colour scientists who develop and evaluate colour correction algorithms and engineers who seek to improve the colour performance of the camera.

This thesis is organised as the following:

In Chapter 2, we set the scene by giving an overview of how colour responses are formed. We do this for both cameras and the HVS. An emphasis is placed on explaining how linear algebra can be used to describe image formation (this helps in later chapters). We also provide a review of the camera processing pipeline.

In Chapter 3, we move on to a discussion of correction algorithms. We review in detail several approaches and empirically evaluate their performance. We also discuss their strengths and weakness from a physical point of view (e.g. whether they work across exposures). In companion to this section we have made a toolbox of algorithms which is available to the community.

In Chapter 4, we provide a discussion of reflectance datasets that have been used in colour correction research. The question of "which dataset should I use" is often asked by researchers. Moreover, the datasets are surprisingly diverse. In this chapter we review existing measures for comparing reflectance datasets, then we introduce a new measure. We then compare reflectance datasets using the measures we described. We show that the choice of reflectance dataset has impact on the performance of colour correction algorithms.

In Chapter 5, we look at the question of reflectance dataset similarity in great depth. We make three important contributions. First, we argue that the surfaces in a reflectance dataset is a subsample of a much larger class of surfaces. Formally, the convex closure of a reflectance dataset is possible, as at micro level, the colours we see are the averages of many distinct surfaces. In our second contribution we show how the convex closure of a reflectance dataset can be represented, and how this in turn can be approximated by its enclosing hypercube. We use the latter for computational necessity, as dealing with high dimensional convex shapes is computationally hard. In our third contribution, we show how we can integrate over our hypercube representation to calculate linear colour correction transforms.

Our experiments also show that the hypercube enclosure of reflectance sets are much more similar than the sampled reflectance dataset themselves. We effectively answer the question of "What data set should I use?" with the answer "It doesn't matter too much so long as we represent reflectances by their enclosing hypercube." Significantly, colour correction driven by the hypercubes delivers colour correction that is almost as good as using the sampled data itself.

In Chapter 6 we consider relaxing the limiting case of the bounding hypercube idea of Chapter 5 by allowing all possible reflectances – any function which is larger than 0% reflectance and less than 100% throughout the visible spectrum. We refer to the resulting maximally large bounding hypercube as Maximum Ignorance with Positivity (MIP). Within the relaxed limiting case, we implement second order polynomial colour correction colour correction algorithms which are more complex than linear regression. This is a theoretical contribution as the resulting transform in practice delivers poor colour correction.

In Chapter 7, we provide a conclusion to this thesis.

Chapter 2

Background

Vision is one of our most important senses, as it informs us about the structure and properties of our physical environment. It is therefore natural for us to want to record what we see. Photography allows us to capture light and create images. It is an art form, as well as a hobby for many people.

Creating images by recording light has been a long running scientific and engineering endeavour. The technology behind photography has evolved over time. The first photograph was taken in 1822 by French Inventor Nicéphore Niṕce [1], using the technique of photoetching. This was the first time in which visible light was recorded as a permanent image. In 1861, photographer Thomas Sutton took a set of three monochromatic images of a tartan ribbon separately, using a camera with red, green and blue filters attached. By projecting these three images to the same location, a colour reproduction of the original tartan ribbon was produced. This idea was first outlined by Scottish physicist James Clerk Maxwell [2].

The 20th century was dominated by film cameras, with consumer grade film cameras becoming more popular in the later half of the century. The first commercially available digital camera was introduced by Sony in 1981 [3]. Since then, digital cameras become increasingly popular. Sharp released the first mobile phone with integrated camera functionality in 2000 [4]. The competition between digital camera and camera phones had been intense in the 2010s. According to the Camera & Imaging Products Association, the sale of digital cameras peaked at 2010. The sales figures has been in gradual decline as due to the improved camera quality of smartphones – consumers prefer to use smartphones with good cameras as their primary camera. As the award winning photographer Chase Jarvis says, “The best camera is the one that’s with you” [5].

Despite the changes in the form factors and the technologies of the cameras, some of the fundamental theories behind creating images remain constant. James Clerk Maxwell used three colour filters to create the three colour channels required for forming colour images.

Colour filters are still present on the imaging sensors of modern digital camera. Three colour channels at the right frequencies are needed for the creation of colour images, due to trichromatic theory of colour vision. Trichromacy of colour vision is due to the physiology of human eyes. The retina at the back of the eye has three types of colour sensitive photoreceptor cells, which are called cone cells.

Colour plays important roles in vision. Colour can provide cues for image segmentation. Primates use colour as a cue for locating fruit in foliage [6]. Colour also provides information regarding the object properties, e.g. identifying the freshness of meat [7] and fish [8], and the ripeness of fruits [9]. Due to the importance of colour in vision, faithful reproduction of colour images is therefore an important topic. However, this problem is complicated by the diversity of imaging devices (which includes cameras and scanners) and the diversity of image reproduction devices (which includes displays and printers).

Horn [10] divided the process of image reproduction grossly into five stages (shown in Figure 2.1). The physical original image is captured by images sensors, a computational subsystem process and store the digital representation of the image, finally an image reproduction device regenerates another image. The regenerated image is finally shown to human observers.

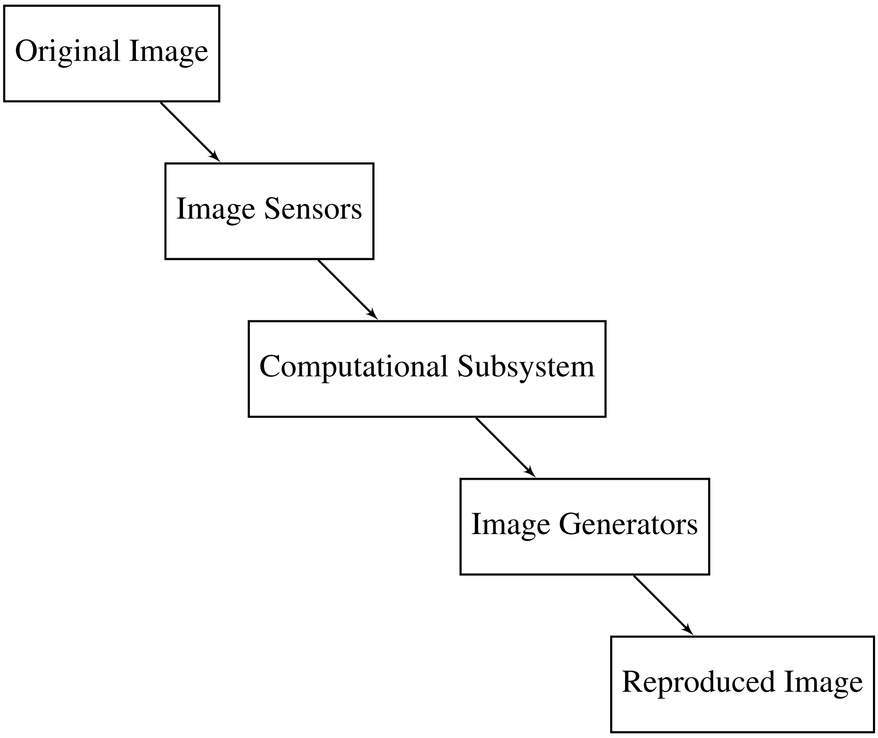


Fig. 2.1 Stages in image reproduction [10]

This thesis concerns the faithful capturing of colour images. In this thesis, we mainly explore the colour correction subcomponent within the "Computational Subsystem" block. However, in this background section, we also touch on the interaction between the "Original Image" and "Image Sensors". We start by exploring the process behind image formation. We then move onto the human visual system (HVS), as the goal of an image reproduction system is to reproduce captured images with that are pleasant to a human observer. We then touch on digital imaging systems, which capture images digitally. Finally we explore why colour correction is necessary.

2.1 Colour Image Formation

In this section, we will discuss the formation of colour signal on imaging sensors. There are different kinds of imaging sensors. In digital cameras, the imaging sensors can be CCD sensors or CMOS sensors. In human eyes, the imaging sensors are the rods and the cones. Despite the fact that different imaging sensors function in different ways due to their physical differences, they can be modelled in a similar manner. Fundamentally, imaging sensors convert incident photons into electrical signals.

A monochromatic imaging sensor can be modelled using the following equation:

$$x = \int_{\omega} E(\lambda)S(\lambda)r(\lambda)\, d\lambda \ , \tag{2.1}$$

where λ is the wavelength. x is the colour response of the imaging sensor, $E(\lambda)$ is the spectral power distribution of electromagnetic radiation which illuminates the scene. $S(\lambda)$ is the reflectance spectra of a surface patch in the scene. $r(\lambda)$ is the device specific spectral sensitivity function of the imaging sensor.

If the imaging sensor consists of m colour channels, we concatenate the responses from each channel to form m-dimensional vector sensor responses $[x_1, x_2, \cdots, x_m]^\intercal$, which we denote as $\mathbf{x}$. We concatenate the specific spectral sensitivity function of each channel together to form a m-dimensional spectral sensitivity vector function $[r_1(\lambda), r_2(\lambda), \cdots, r_m(\lambda)]^\intercal$, which we denote as $\mathbf{r}(\lambda)$. By substituting $\mathbf{x}$ and $\mathbf{r}(\lambda)$ into Equation 2.1, we obtain the sensor response for a multi-channel imaging sensor:

$$\mathbf{x} = \int_{\omega} E(\lambda)S(\lambda)\boldsymbol{r}(\lambda)\, d\lambda \ , \tag{2.2}$$

When we are dealing with the HVS, we replace $\mathbf{r}(\lambda)$ with the CIE colour matching functions [11], which is the spectral sensitivity function for the HVS. By using the CIE colour matching function, we can calculate the XYZ tristimulus values, which numerically describes the colour response of a typical human observer (in the context of a colour matching experiment). Since colour matching functions are linearly related to cone responses they are taken as a measure of what we see (in terms of the signal measured by the sensors in the eye). CIE colour matching functions are discussed further in a later subsection below.

It should be noted that a typical imaging device tends to have 3 colour channels $(m = 3)$. This is because human eyes have 3 different types of cones. However this is not always the case, for example, hyperspectral cameras may have more than 3 colour channels in order to cover electromagnetic radiation outside the visible range, such as infrared radiation or ultraviolet radiation [12].

By multiplication of the illuminant $E(\lambda)$ and the reflectance $S(\lambda)$ together, we can obtain the actual colour signal ($C(\lambda)$) that reaches the imaging sensor:

$$C(\lambda) = E(\lambda)S(\lambda) \; . \tag{2.3}$$

By substituting Equation 2.3 into Equation 2.2, it can be simplified into:

$$\mathbf{x} = \int_{\omega} C(\lambda)\mathbf{r}(\lambda)\; d\lambda \; , \tag{2.4}$$

In Equation 2.4, the colour formation model is presented in continuous form. This is more realistic in terms of modelling the physical process. However, it is reasonable to discretise the colour signal and device specific spectral sensitivity function. Discretisation of continuous distribution is carried out by taking samples at regular intervals.

The visible spectrum is the range of electromagnetic wave that a human eye will respond to. The exact range varies between individuals, and it is between about 365 nm to 750 nm [13]. However in context of computer vision, only the wavelength between 400 nm and 700 nm are taken into account, as the responses produced by stimuli below 400 nm and above 700 nm in human observers are fairly weak.

2.1.1 Discrete Representation

In practice, computation of colour responses is done in discrete domain rather than continuous domain. This is because spectrum data mentioned in Equation 2.4 can only be sampled in discrete intervals. By sampling the colour signal $C(\lambda)$ and spectral sensitivity function $\mathbf{r}(\lambda)$ at a regular 10 nm interval between 400 nm to 700 nm, we can represent $\mathbf{r}(\lambda)$ using a $31 \times m$ matrix $\mathbf{R}$ ($m = 3$ for RGB imaging sensor), and $C(\lambda)$ can be represented using a 31-column vector **c**. In this way, we have that,

$$\lambda_i = 400 + 10i \ (i = 1..31) \tag{2.5a}$$

$$(\mathbf{R})_{ji} = \mathbf{r}_j(\lambda_i) \tag{2.5b}$$

$$(\mathbf{c})_i = C(\lambda_i)\Delta_i \ , \tag{2.5c}$$

where Δ_i is the sampling interval.

By adopting the convention above and assuming sampling interval is incorporated in the vector approximation, Equation 2.4 can be replaced by matrix multiplication:

$$\mathbf{x} = \mathbf{R}^{\mathsf{T}}\mathbf{c} \tag{2.6}$$

2.1.2 Colour spaces

A colour space is a distinct representation of colour. As our understandings of colour evolve over time, different representations of colour have been developed as different interest groups utilise colour in different ways. In the previous section, we discussed that the device specific spectral sensitivity function is used when calculating the colour response of a digital camera, while the CIE colour matching functions is used to calculate the XYZ tristimulus of HVS. Due to the fact that in general, the device specific spectral sensitivity functions are different to the CIE colour matching functions, digital cameras produce different responses than the HVS for the same colour stimulus. The responses from a digital camera occupy a different colour space to the responses from HVS. This is partly due to the engineering difficulty associated with imaging sensor design and noise property constraints of the signal (especially when interactions between imaging and display devices are involved).

Device independent colour spaces can be broadly classified into two categories: unrendered colour space, and rendered colour space [14]. Rendered colour spaces are designed with being used for device output purposes in mind. They tend to be smaller than unrendered colour spaces, and therefore in general, smaller ranges of colour can be expressed in rendered colour spaces compared to unrendered colour spaces. Unrendered colour spaces are not designed for output purposes, they are designed for storage and calculation. Colours which cannot by physically realised by output devices can be represented in unrendered colour spaces. Examples of this type of colour space include CIE XYZ [15], RIMM RGB [16] and etc. Rendered colour space are designed for output purposes. Example of rendered colour space include ITU-R BT.709-3, sRGB [17],and etc.

2.2 Human Visual System

The Human Visual System (HVS) consists of various components, however the most important two parts of are the eyes and the visual cortex. The eyes convert the received light into electrical signal. The electrical signal is then transmitted to visual cortex via the optic nerve. The visual cortex process and interpret the received electrical signal.

Figure 2.2 shows a schematic diagram of a human eye. Light first enters the eye through the pupil. The pupil controls the amount of light that enters the eye by adjusting its diameter. In a bright environment, the pupil constricts to reduce the amount of light that enters . In a dark environment, the pupil dilates to allow more light to enter. The light is then focused by the lens onto the retina. The curvature of the lens can be changed by the ciliary muscle, this allows objects from different distances to come into focus. The retina is the light sensitive tissue responsible for converting light energy to electrical neural impulses.

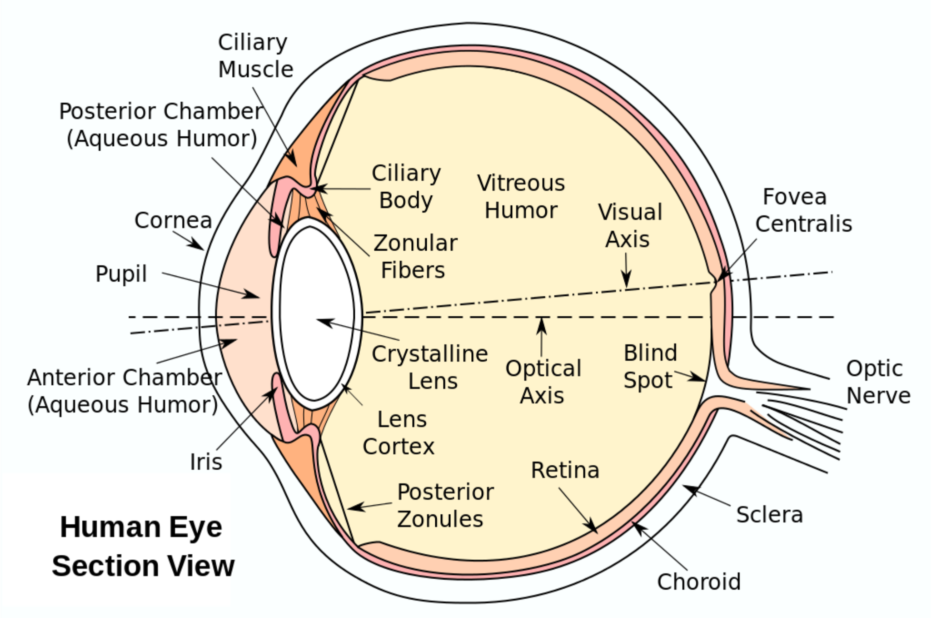


Fig. 2.2 The schematic diagram of the human eye, the diagram has been released into the public domain by the original author [18]

The retina consists of layers of neurones and capillaries, and a layer of photoreceptor cells. In human (and other vertebrates), light has to pass through layers of neurones and capillaries before reaching the photoreceptor cells. An image of section of the author's retina imaged using Adaptive Optics Scanning Laser Ophthalmoscopy [19] is shown in Figure 2.3. The grains in the image are the cone cells, the dark patch in the middle of the image is a blood vessel. The image shows that cone cells are tightly packed together. This enables a high degree of visual acuity.

Fig. 2.3 The author's retina imaged using Adaptive Optics Scanning Laser Ophthalmoscopy (AOSLO) at Department of Experimental Psychology, University of Oxford. The grains in the image are cone cells, the dark patch in the middle of the image is a blood vessel.

There are two types photoreceptor cells in human retina – rods and cones. Two types of photoreceptor cells are responsible for transducing light to electrical. The type of cells used depends on ambient illumination level. At very low light level, rods are responsible for the monochromatic vision that we experience. This is referred to as scotopic vision. In most night time outdoor and street light condition, human vision is in the mesopic range [20]. In mesopic vision, a combination of cones and rods are responsible for vision. When the environment is illuminated to day light level, cones alone are responsible for vision. This is referred to as photopic vision. Most of the work on colour correction focus on photopic vision.

There are three types of cones. They are named based on the wavelength of the visible light they respond to – *L* (long) cones have peak sensitivity near 564-580 nm, for *M* (medium) it is around 534-545nm, and for *S* (short), it is around 420-440 nm [21]. *S* cones appear randomly and far less frequently than *M* and *L* cones. The ratio of *M* and *L* cones varies greatly among individuals with normal colour vision [22]. The average sensitivities for the three types of cones is show in Figure 2.4 [23]. This figure describes how much neural impulse each type of cone cell produces given a monochromatic light stimulus at a certain frequency. The measurement of cone sensitivities is beyond the scope of this thesis. The CIE XYZ colour matching functions are related to the cone sensitivities, as they are linear combinations of the cone sensitivities.

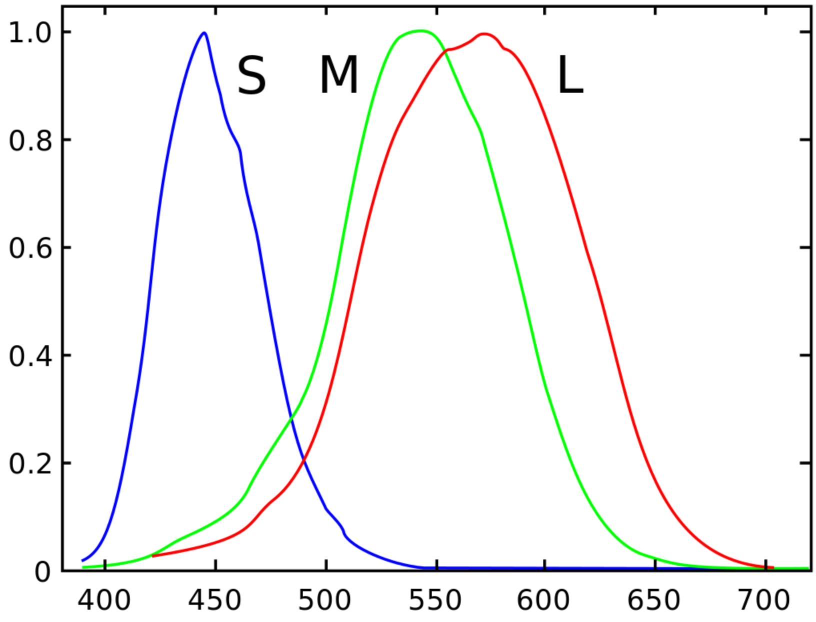


Fig. 2.4 The sensitivities of human cone [23]

2.2.1 Trichromatic Theory of Colour Vision

Photopic vision – the vision of the eye under well-lit condition, can be best explained using trichromatic theory of colour vision. This is primarily because the three types of cones are responsible for this mode of vision. The development of trichromatic theory began as scientists investigated what makes up visible light.

Young [24] observed that painters mix red, yellow and blue pigment to achieve a wide variety of colour. Based on these observation, Young postulated that visible light comes in a continuum of frequencies, but the human eyes have only three types of receptors. He believed that these receptors responds to red, yellow and blue light.

Maxwell studied the composition of colour using a wooden spinning top [25, 26]. An illustration of Maxwell's spinning top is shown in Figure 2.5. A colour sample was placed at the middle of the spinning top. The outer rim of the spinning top consists of three pieces paper coloured using the primary colour red, green and blue. Maxwell performed experiments by spinning the top at a high speed, and due to the persistence of vision, the colour in the outer rim would blend into one colour. Maxwell demonstrated that by varying the amount of primary colour at the outer rim, it is possible for the outer rim's blended colour to match the colour of the sample at the middle of the spinning top. He hypothesised that all colour could be synthesized by varying the amount of light with primary colour.

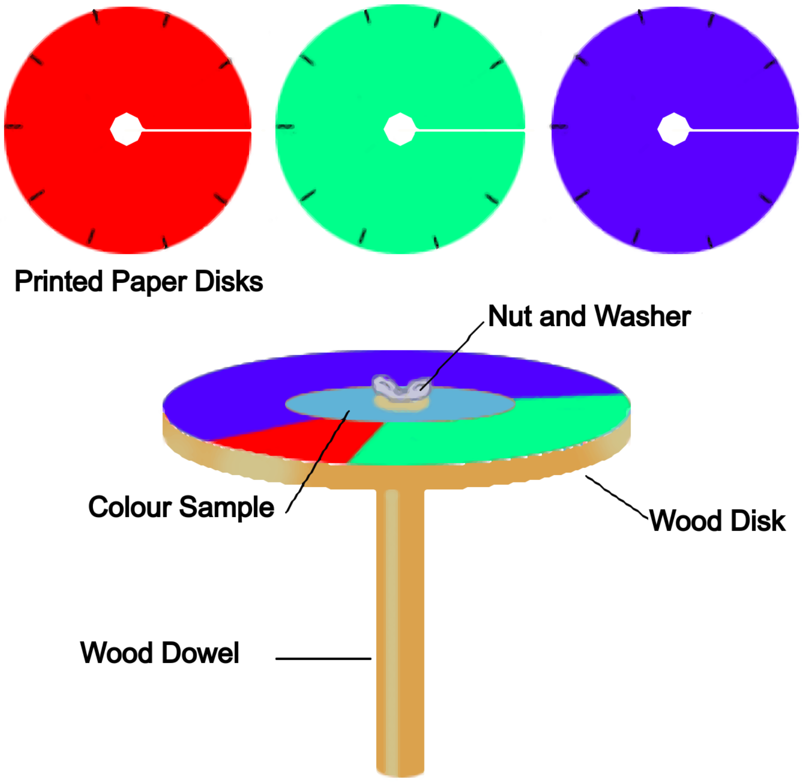


Fig. 2.5 An illustration of Maxwell's wooden colour spinning top for his original colour matching experiment

Maxwell discovered the difference between additive colour mixing and subtractive colour mixing models. In both situations, new colours forms as the result of light of different colours mixing together. However in additive colour mixing, one starts with black, which is the absence of light. As lights of different wavelength mix together, colour forms. In subtractive colour mixing, one starts with a "white" surface, which is a surface which reflects of the illuminant equally across different wavelengths. By adding pigment to the surface, light of certain wavelengths are subtracted, which results in colour. The colour model behind cameras, eyes and displays is additive colour mixing. The colour model used by printers is subtractive colour mixing.

The spinning top experiment performed by Maxwell was effectively a “colour matching experiment”, in which the participant attempts recreate a sample colour by mixing varying amount of light with primary colours.

2.2.2 CIE Primaries

CIE stands for Commission Internationale de l’Éclairage. In English, it means the International Commission on Illumination. It is the international organisation that sets standards on illumination and colour. CIE took the idea of colour matching experiment further, by defining what primary colours are, and created various associated colour spaces. In particular, CIE combined the work from two separate teams, Wright [27] from Imperical College London, and Guild [28] from the National Physical Laboratory.

Both researchers conducted colour matching experiments using participants with normal colour vision. The participants were asked to look into a device called “colorimeter”, which had a square field of view divided horizontally into two equal rectangles. Each side of the square occupied 2°of vision. This ensured that only the fovea was involved in the colour matching. One of the rectangle was the test field, which showed a monochromatic light – it was the colour to be matched. The other rectangle was the match field, it showed the result of three primary colours blended together. Each primary colour was a monochromatic light – its spectrum contained one sharp peak at a single wavelength.

The participants were asked to adjust the strength of the primaries, so the colour in the match field matched the colour in the test field. The strengths of the primaries needed to produce those colour responses were recorded, these are known as the *colour matching functions*. A diagram representing the setup of the colour matching experiment is shown in Figure 2.6.

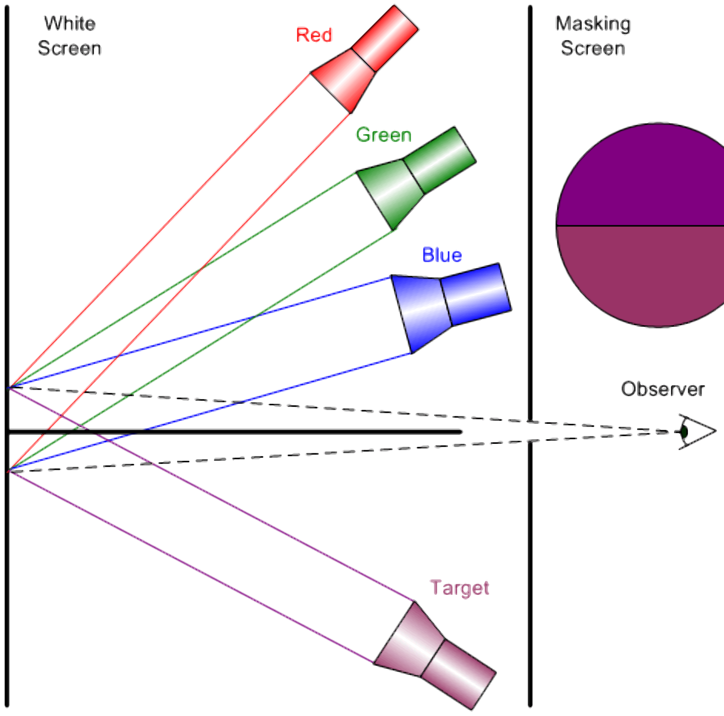


Fig. 2.6 A diagram representing the colour matching experiment, the participants adjust the strength of the red, green and blue light beam, so when these three light beams mix, the resulting colour matches the colour of the test source, image taken from [29]

The two team used two different primary colours, as they used different methods to isolate those monochromatic lights, Guild [28] used a tungsten lamp and colour filters, while Wright [27] used a system of prisms. CIE committee combined the results from these two research groups using mathematical transformation, resulting in CIE 1931 RGB Colour Matching Functions. The primaries were standardised to 700 nm, 546.1 nm and 435.8 nm, and they are denoted using $\bar{r}(\lambda)$, $\bar{g}(\lambda)$ and $\bar{b}(\lambda)$ respectively. CIE 1931 RGB Colour Matching Functions are shown in Figure 2.7.

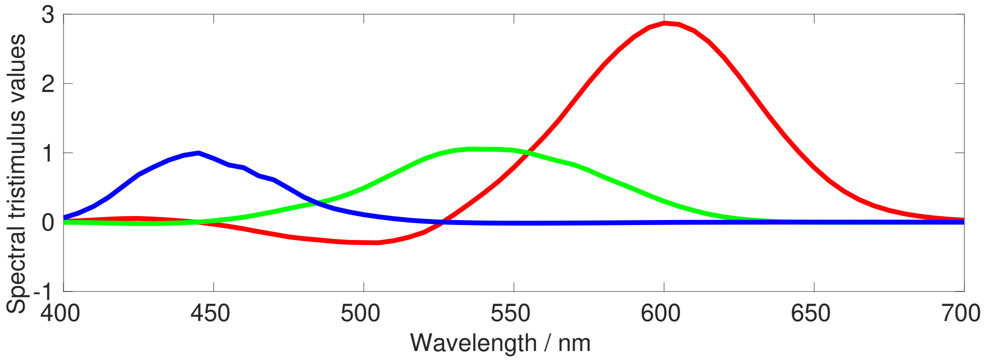


Fig. 2.7 The CIE 1931 RGB Colour Matching Functions. The red curve is $\bar{r}(\lambda)$, the green curve is $\bar{g}(\lambda)$ and the blue curve is $\bar{b}(\lambda)$.

During colour matching experiments, there were spectral colours which could not be matched by mixing primaries in the *match* field. The solution was to mix the red primary to into the *test* field. This solution is represented mathematically using negative values in $\bar{r}(\lambda)$. This effect is especially prominent at around 500 nm.

2.2.3 CIE XYZ colour matching functions

The CIE commission felt that the negative values in the RGB colour matching function might lead to error in its usage and impede its adoption. In order to avoid the negative values, CIE devised the CIE XYZ colour matching function.

The CIE XYZ colour matching functions are denoted using the symbol $\bar{x}(\lambda)$, $\bar{y}(\lambda)$ and $\bar{z}(\lambda)$. The CIE XYZ colour matching functions are a linear combination of the CIE RGB colour matching function. It is designed in such a way so that negative numbers do not exist in the colour space. Additionally, the Y values in CIE XYZ are proportional to the overall luminance of a colour [30].

The CIE XYZ colour matching functions can be calculated from the CIE RGB colour matching functions using the following formulae [15]:

$$\begin{aligned}
\bar{x}(\lambda) &= 0.49\bar{r}(\lambda) + 0.31\bar{g}(\lambda) + 0.20\bar{b}(\lambda) \\
\bar{y}(\lambda) &= 0.17697\bar{r}(\lambda) + 0.81240\bar{g}(\lambda) + 0.01063\bar{b}(\lambda) \\
\bar{z}(\lambda) &= 0.00\bar{r}(\lambda) + 0.01\bar{g}(\lambda) + 0.99\bar{b}(\lambda)
\end{aligned} \tag{2.7}$$

The CIE XYZ colour matching function is shown in Figure 2.8.

The CIE XYZ colour space can represent all possible colour sensation an average person can perceive, therefore it is used as a device independent colour space. Mapping from device dependent colour space to CIE XYZ is an important tool when designing and analysing the

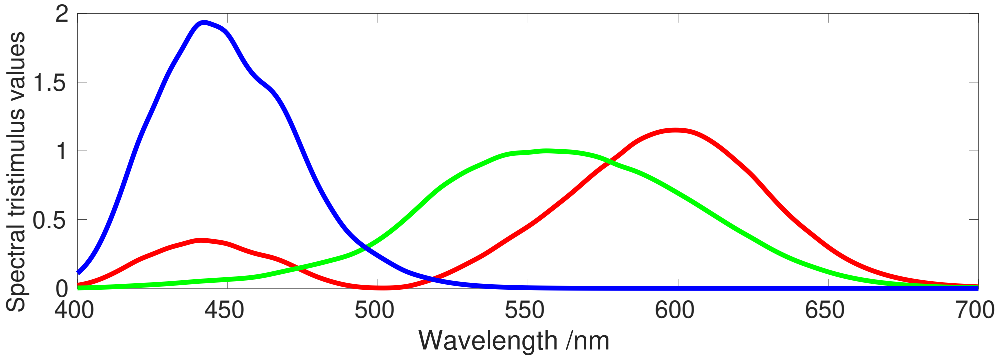


Fig. 2.8 The CIE XYZ colour matching function, The red curve is $\bar{x}(\lambda)$, the green curve is $\bar{y}(\lambda)$ and the blue curve is $\bar{z}(\lambda)$.n

image processing pipeline for cameras [14], although the cameras themselves may never explicitly use this colour space. When we are analysing how good a camera is at measuring colour, we are typically interested in how well a camera's device dependent RGBs can be corrected to the corresponding XYZs. We measure the difference between the corrected XYZs as reported by the camera and the true XYZs measured by another more accurate colorimetric device, e.g. a spectrophotometer. The smaller the differences are, the more colour accurate the camera is.

The CIE XYZ colour space cannot be used directly to measure the perceptual difference between two colours, as it is not perceptually uniform [31]. In a perceptually uniform colour space, the same numerical change in the colour values must correspond to the same magnitude of perceived change in colour by human observers across the whole colour space [21]. This is not quite the case with CIE XYZ. In particular, the same numerical differences result in a much larger magnitude of perceptual difference in the green region than the blue region. This effect was first investigated by MacAdam [32].

Perceptually uniform colour spaces have been derived from CIE XYZ colour space. The most prominent ones are CIELAB and CIELUV [33], CIELAB is the more commonly used.

2.2.4 CIELAB colour space

CIELAB colour space is a approximately perceptually uniform colour space based on CIE XYZ. In this colour space, it is easy to calculate the perceptual differences between two colours.

CIE XYZ can be converted to CIELAB using the folllowing formulae [34]:

$$\begin{aligned}
L^* &= 116f\left(\frac{Y}{Y_n}\right) - 16 \\
a^* &= 500(f\left(\frac{X}{X_n}\right) - f\left(\frac{Y}{Y_n}\right) \\
b^* &= 200(f\left(\frac{Y}{Y_n}\right) - f\left(\frac{Z}{Z_n}\right))
\end{aligned} \tag{2.8}$$

, where

$$f(t) = \begin{cases} \sqrt[3]{t} & : t > \left(\frac{6}{29}\right)^3 \\ \frac{841}{108}t + \frac{4}{29} & : t \leqslant \left(\frac{6}{29}\right)^3 \end{cases}, \tag{2.9}$$

and X_n, Y_n and Z_n are the XYZ tristimulus values of the whitepoint.

The '$L*$' in CIELAB is the lightness factor, which measures the brightness of the colour on a scale between 0 to 100, with 0 being the darkest black, and 100 being the brightest white. The 'a*' measures the ratio of red and green, with positive values representing red, and negative values representing green. The 'b*' measures the ratio of blue and yellow, with positive values representing yellow, and negative values representing blue [35].

In order to calculate the perceptual difference between two colours in CIELAB colour space, the colour difference formulae need be used. These formulae have evolved over time.

CIE76 Colour Difference Formula

CIE76 is the original colour difference formula. It is is defined as [33]:

$$\Delta E^*_{ab} = \sqrt{(\Delta L^*)^2 + (\Delta a^*)^2 + (\Delta b^*)^2}. \tag{2.10}$$

This formula calculates Euclidean distance between two colour coordinates within the CIELAB colour space. The result is known as the CIELAB ΔE^*_{ab} , also commonly referred to as the "CIELAB difference". The CIELAB ΔE^*_{ab} is typically used for evaluating the performance of colour correction algorithm. In terms of fine human colour difference judgements, a ΔE^*_{ab} of 1 roughly corresponds a difference that's perceptually just noticeable.

CIE94 Colour Difference Formula

The CIELAB colour space is only *approximately* perceptually uniform. It was indicated that the use of different weighting for lightness, chroma and hue differences may be necessary in different practical application. Various formulae have gradually evolved to address these issues. A prominent example was the CIE94 formula released in 1994 [36]. It provides significant improvement in accuracy over CIELAB [37].

The CIE94 colour-difference formula is designated as $\Delta E^*_{94}(k_L{:}k_C{:}k_H)$, it is defined as:

$$\Delta E^*_{94}(k_L{:}k_C{:}k_H) = \sqrt{\left(\frac{\Delta L^*}{k_L S_L}\right)^2 + \left(\frac{\Delta C^*_{ab}}{k_C S_C}\right)^2 + \left(\frac{\Delta H^*_{ab}}{k_H S_H}\right)^2}, \tag{2.11}$$

where the CIELAB lightness (ΔL^*), chroma (ΔC^*_{ab}), and hue differences(ΔH^*_{ab}) are given by:

$$\Delta L^* = L^*_1 - L^* - 2 \tag{2.12}$$

$$\Delta C^*_{ab} = C^*_{ab,1} - C^*_{ab,2} \tag{2.13}$$

$$\Delta H^*_{ab} = 2\sqrt{C^*_{ab,1} C^*_{ab,2}} \sin\left(\frac{\Delta h_{ab}}{2}\right) \tag{2.14}$$

$$\Delta h_{ab} = h_{ab,1} - h_{ab,2}. \tag{2.15}$$

The “weighting functions” for lightness (S_L), chroma (S_C) and hue (S_H) are defined as:

$$S_L = 1 \tag{2.16}$$

$$S_C = 1 + 0.045(C^*_{ab,1} C^*_{ab,2}) \tag{2.17}$$

$$S_H = 1 + 0.015(C^*_{ab,1} C^*_{ab,2}). \tag{2.18}$$

The “parametric factors k_L, k_C, k_H are set to 1.0. However in the textile industry, it is a common practice to set the lightness parametric factor to 2.0.

CIEDE2000 Colour Difference Formula

In 2001, the CIE released a new colour difference formula - CIEDE2000[38]. It provides the improved overall accuracy compared to CIE94, by further correcting perceptual non-uniformity within the CIELAB colour space.

The CIEDE2000 colour difference is designated as $\Delta_i 00$, it is calculated using a three-step process described below.

Step 1: Preparation of data to calculate a', C' and h':

$$L' = L^* \tag{2.19}$$

$$a' = (1+G)a^* \tag{2.20}$$

$$b' = b^* \tag{2.21}$$

$$C'_{ab} = \sqrt{a'^2 + b'^2} \tag{2.22}$$

$$h_{ab} = tan^{-1}\left(\frac{b'}{a'}\right) \tag{2.23}$$

where

$$G = 0.5\left(1 - \sqrt{\frac{\overline{C^*_{ab}}^7}{\overline{C^*_{ab}}^7 + 25^7}}\right), \tag{2.24}$$

and $\overline{C^*_{ab}}$ is the arithmetic mean of C^*_{ab} values for a pair of samples.

Step 2: Calculate $\Delta L', \Delta C'$ and $\Delta H'$

$$\Delta L' = L'_2 - L'_1 \tag{2.25}$$

$$\Delta C'_{ab} = C'_{ab,2} - C'_{ab,1} \tag{2.26}$$

$$\Delta H'_{ab} = 2\sqrt{C'_{ab,2}, C'_{ab,1}} \sin\frac{\Delta h'_{ab}}{2} \tag{2.27}$$

where

$$\Delta h'_{ab} = h'_{ab,2} - h'_{ab,1} \tag{2.28}$$

Step 3: Calculate CIEDE2000 ΔE_{00}

$$\Delta_{00} = \sqrt{\left(\frac{\Delta L'}{k_L S_L}\right)^2 + \left(\frac{\Delta C'_{ab}}{k_S S_C}\right)^2 + \left(\frac{\Delta H'_{ab}}{k_H S_H}\right)^2 + R_T\left(\frac{\Delta C'_{ab}}{k_C S_C}\right)\left(\frac{\Delta H'_{ab}}{k_H S_H}\right)}, \tag{2.29}$$

where

$$S_L = 1 + \frac{0.015(\bar{L}' - 50)^2}{\sqrt{20 + (\bar{L}' - 50)^2}} \tag{2.30}$$

$$S_C = 1 + 0.045\overline{C'_{ab}} \tag{2.31}$$

$$S_H = 1 + 0.045\overline{C'_{ab}}T \tag{2.32}$$

$$T = 1 - 0.17\cos\left(\overline{h'_{ab}} - 30^\circ\right) + 0.24\cos\left(2\overline{h'_{ab}}\right) + 0.32\cos\left(3\overline{h'_{ab}} + 6^\circ\right) + \tag{2.33}$$

$$0.20\cos\left(4\overline{h'_{ab}} + 63^\circ\right) \tag{2.34}$$

$$R_T = -\sin(2\Delta\theta)R_C \tag{2.35}$$

$$\Delta\theta = 30\exp\left\{-\left[\frac{\overline{h'_{ab}} - 275^\circ}{25}\right]^2\right\} \tag{2.36}$$

$$R_C = 2\sqrt{\frac{7\overline{C}_{ab}}{7\overline{C}_{ab} + 25^7}} \tag{2.37}$$

Note that $\overline{L'}$, $\overline{C'_{ab}}$ and $\overline{h'_{ab}}$ are arithmetic means of the L', C'_{ab} and h'_{ab} values for a pair of samples.

2.3 Digital imaging system

Digital cameras capture images by converting light signals into electrical signals using imaging sensors [39]. The typical processing pipeline of a digital camera is shown in Figure 2.9 [14]. A typical digital camera has all the blocks in the flow chart, however the implementation of the same blocks may be very different between cameras may be very different, as manufacturers tune their cameras differently. We now explore a few selected blocks within the flowchart.

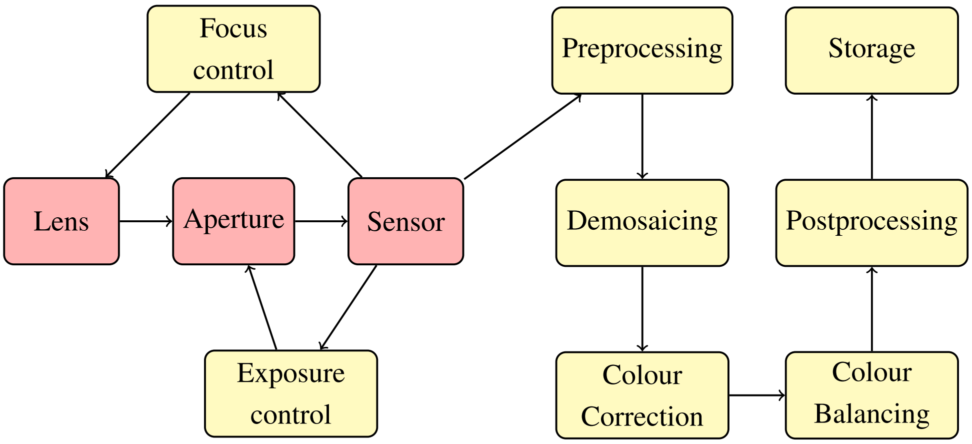


Fig. 2.9 A typical camera image processing pipeline [14], the red blocks represent physical components, while the yellow blocks represent processes. It should be noted that the exact processing sequence differs between manufacturers.

2.3.1 Focus control

Focus control is primarily achieved by two methods - contrast measurement and phase detection. In contrast measurement, the image is first divided into different regions. The lens system is then adjusted so the contrast and the amount of high spatial frequency content in the foreground region is maximised [14]. In camera systems that use phase detection, the autofocus sensor is effectively a rangefinder. The incoming light is split into two using a beam splitter. This creates two images, and their intensity patterns are then compared and analysed. The phase difference between the two images is then used to calculate the focus with respect to the scene [40].

2.3.2 Exposure control

Exposure controls the amount of light recorded by the imaging sensor. Overexposure happens if the imaging sensor capture too much light. Overexposed images contain low contrast bright regions lacking in spatial details which are indistinguishable from white. Underexposure happens if the imaging sensor does not capture enough light. Underexposed images contain low contrast dark regions lacking in spatial details which are indistinguishable from black.

There are two ways to achieve the control of the level of exposure - the camera can change the size of the aperture and/or change the shutter speed. It should be noted that on cheaper cameras, the size of the aperture may be fixed. However, the shutter speed can be changed on all cameras.

Scene with high dynamic range presents a particular challenge for digital camera. Dynamic range refers to the contrast ratio between the brightest pixel and the darkest pixel within an image. The HVS is capable of distinguishing contrast over a range of 3.7 orders of magnitude [41], while a typical camera and a typical display has a dynamic range of 2 orders of magnitude [42]. This means that for an image of the scene with extreme contrast, overexposed region and underexposed region may occur at the same time. This is because the camera can only record a limited range of the full contrast. The solution for this problem is through the use of high dynamic range (HDR) imaging [43]. HDR images are created by fusing multiple images of the same scene captured different exposure level. The creation of HDR image is beyond the scope of this thesis, more information can be found in [43].

2.3.3 Sensor

After the focus and exposure are set, the imaging sensor can finally start to capture the scene. There are two types of imaging sensor - Charged Coupled Device (CCD) and Complementary Metal Oxide Semiconductor (CMOS). The details on how these sensors operate is beyond the scope of this thesis, more information can be found in [39].

The main advantage for CMOS sensors is that they tend to be cheaper than CCD sensor. This leads to their wide adoption in the low end market, traditionally webcams and mobile phone cameras use CMOS sensors. Historically, CCD sensors is considered as a more mature technology. CCD were able to produce images with lower noise compared to CMOS sensors [44]. However due to improvement in CMOS sensor technology, this is no longer the case. Since 2010s, DSLR cameras have been exclusively using CMOS sensors [45].

The major difference between CMOS sensors and CCD sensors is the way the electronic shutter operates. CMOS sensors use a rolling shutter, while CCD sensor uses a global shutter. When taking picture using a camera with a CMOS sensor, the scene is not actually captured instantaneously at once. The scene is captured by recording pixel values from the imaging sensor one-by-one from each row, then row-by-row. This is similar to the way the electron beam from a CRT display scan across the screen. Rolling shutter can produce imaging artefacts. Figure 2.10 shows spatial aliasing artefact. This artefact occurs when the shutter speed is below the sampling frequency required for the subject.

Fig. 2.10 An image showing the effect of rolling shutter, with the red rectangle encircles the affected region. The shape of the propeller blade of the helicopter is physically different to what is shown in the image - it should be straight. This image is licensed under the Creative Commons Attribution-Share Alike 3.0 Unported license [46]

2.3.4 Colour filter array and demosaicing

For a camera to form colour images, information for three colour channels need to be generated at each pixel location. There are primarily two ways to achieve this – using a colour filter array (CFA) on one single sensor then use interpolation, or using three imaging sensors to measure each channel independently. By using a CFA, each pixel location on the imaging sensor measures one single colour channel, the information of the two missing colour channels are interpolated from the adjacent pixels. Having multiple sensors in a single camera means that information about all three colour channels are available for each pixel location. Having multiple sensors produce images with better quality, However this greatly increases the size of the camera, as well as the cost. Therefore, most consumer grade digital cameras have one single imaging sensor with a CFA.

There are many CFA patterns available. However the most popular CFA is the Bayer array [47]. Bayer array consists of one red mask, one blue mask and two green masks. The Bayer array patten is shown in Figure 2.11. Camera manufacturers have produced cameras using different CFA pattern, however they are not very common.

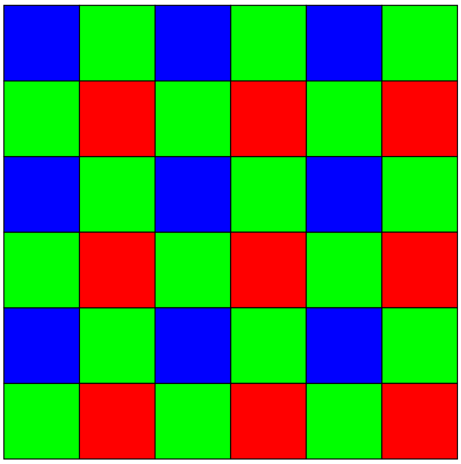


Fig. 2.11 A diagram showing a Bayer pattern array which contains three sets of repeated pattern both horizontally and vertically.

The use of CFA means that at a single pixel location only the intensity value for one single colour channel is measured. To fill in the information for other channels, a computationally intensive process called “demosaicing” is required. Different cameras use different demosaicing algorithm. Quite often the exact implementation is covered by patents or proprietary. There is a large volume of researches on demosaicing, [48, 49] provide good reviews in this topic. Popular demosaicing techniques include weighted sum interpolation [50], adaptive homogeneity-directed demosaicing [51], succesive approximation [52], Bayesian approaches [53], neural network [54, 55] and etc. Poorly designed demosaicing algorithm tends to introduce artefacts into the resulting image, i.e. the zipper artefacts [52] and confetti artefacts [56].

2.3.5 Preprocessing

After the imaging system receives a command to capture an image, a record of the intensity values from each pixel location on the imaging sensor is created. We refer to this record as the “raw” image. The first stage of the image processing pipeline is the preprocessing of the raw image. This involves the following processes: removal of readings from defective pixels (due to imperfect manufacturing processes) [57], linearisation of the camera responses using an opto-electronic conversion function [58, 59], compensation of the dark current (the thermal noise inherent to the imaging sensor) [60], compensation of the flare caused by the undesired scattering and reflection of light by the components in the optical system [61]. The purpose of preprocessing is to improve the performance of the downstream stages within the pipeline.

2.3.6 Colour correction

Cameras measure colour differently to the HVS, due to the spectral sensitivity difference between the imaging sensors of the cameras and the cone cells of human eyes. To faithfully reproduce colours as perceived by the HVS, it is therefore necessary to convert the device specific colour responses produced by the camera to a well-defined device independent colour spaces linked to the HVS. This process is called colour correction.

Colour correction in digital a digital camera conceptually can be thought of as a two step process. The raw sensor reading is first transformed from the device specific colour space to an unrendered colour space, then from that unrendered colour space to a rendered colour space [14]. However, quite often the two operations are combined together mathematically, so colour correction can be completed in one step. However, since colour corrections are generally evaluated with respect to an unrendered space (typically XYZ) then it is useful to think of the process as being in two stages.

As colour correction is the main topic of this thesis, we devote Chapter 3 to the introductory discussion of this topic.

2.3.7 Colour balancing

Colour balancing is the global adjustment of the colours, so the objects in the image appear to have the expected colour. Colour balancing is necessary, because the illuminant can influence the colour appearance of the object. To understand the reasoning behind this, we need to revisit the equation that describe the formation of imaging sensor responses. According to Equation 2.4, if the surface reflectance $S(\lambda)$ and the device spectral sensitivity function $\mathbf{r}(\lambda)$ are held constant, a changed illuminant $E(\lambda)$ results in a different sensor response $\mathbf{x}$.

The HVS has the natural ability to take the illuminant into account when sensing colour. Despite that a sheet of white paper generate different XYZ tristimulus values under different illuminants (e.g a 3600K lightbulb and daylight), the HVS interprets the colour of the paper to be white as long as the illuminant has a relatively neutral colour. This phenomenon is known as colour constancy [62]. This behaviour is not inherent to a machine vision system. The ability to reduce and remove the influence of the illuminant to the final image needs to be programmed into the image processing pipeline. An important goal of colour balancing is to adjust the appearance of neutral colour, especially white. Therefore colour balancing is also often referred to as "*white balancing*".

There are various algorithms to perform colour balancing, with the end goal of achieving colour constancy for a machine vision system. The simplest approaches are Grey World algorithm Buchsbaum [63], which makes the assumption that the average pixel colour over the entire image is grey. Each colour channel is scaled by the mean intensity value of that channel, i.e:

$$I_c(x,y) = \frac{I'_c(x,y)}{\mathrm{mean}(I'_c)}, \tag{2.38}$$

where $I_c(x,y)$ is the scaled pixel intensity at coordinate (x,y), and $I'_c(x,y)$ is the corresponding unscaled pixel intensity and $\mathrm{mean}(I'_c)$ is the mean pixel intensity of colour channel c.

Scale by Max algorithms [64] each colour channel is scaled by the maximum intensity value of that channel, i.e:

$$I_c(x,y) = \frac{I'_c(x,y)}{\max(I'_c)}, \tag{2.39}$$

Both Grey World algorithm and Scale by Max algorithms are naive approaches which are simple to implement, however and they perform poorly on colourful images, which do not necessarily fit with the assumption of these algorithms. They suffer from instability on video footage with dynamic scene,

More advanced approaches of colour balancing includes the Retinex algorithm by Land and McCann [65], which is based on the psychological aspects of lightness and colour perception of human vision. There are other algorithms which are improvements over Retinex algorithm, e.g. Hurlbert [66] proposed a Gaussian surround function, Multiscale Retinex [67] which incoporates a number of image processing operations. More information about white balancing and colour constancy can be found at [68–72].

2.3.8 Postprocessing

The goal of postprocessing is to remove the artefacts generated in the previous processing steps, and to enhance the final appearance of the image. Common postprocessing steps include colour artifact removal, edge enhancement and tone mapping.

Colour artifact removal refers to the process of removing the zipper artefacts and confetti artifacts introduced during the demosaicing step [73]. These artefacts need to be removed as they are visually objectionable. The exact process in this step differs between camera models, and they tend to be proprietary.

Human visual system is highly sensitive to sharp edges [74]. Experiments have shown that observers tend to prefer sharp edges over blurred edges [75]. In order to make images more aesthetically pleasing, most camera manufactuers apply edge enhancement during postprocessing. A commonly used edge enhancement technique is unsharp masking [76].

Tone mapping is one of the final step in postprocessing. Typically, the imaging sensor of the camera has a higher dynamic range than the output file format (typically JPEG). Tone mapping allows the high dynamic range intermediate image to be mapped down to a low dynamic range storage format, which remain aesthetically pleasing [77]. For producing a standard low dynamic range image, tone mapping is a global operator – the same tone mapping function is applied across the whole image. However to produce HDR image, spatially varying local tone mapping is necessary [78].

2.3.9 Storage

Generally there are primarily two ways for the camera to store the image taken: JPEG/Exif format and device specific raw image format.

JPEG/Exif stands for Joint Photographic Experts Group/Exchangeable Image Format. Joint Photographic Experts Group (JPEG) developed the compression standard for this file format [79]. However the file format itself is jointly maintained by Japan Electronics and Information Technology Industries Association (JEITA) and Camera & Imaging Products Association (CIPA) [80]. JPEG/Exif files have the extension of '.JPG'. JPEG compression standard is a lossy form of compression utilising discrete cosine transform (DCT). The image is first converted into the frequency domain. The compression is achieved by discarding high frequency information based on a psychovisual perceptual model. Storing fully processed image in JPEG/Exif format is available for all digital still cameras.

High end cameras such as DSLR cameras, and more recently mobile phones [81, 82] can also store device specific raw images. A raw image is the unprocessed readings captured by the imaging sensor – the captured image data before going through the "preprocessing" step in Figure 2.9. The exact format of the raw image is dependent on the exact model camera. There is no industry-wide agreement on the format of the raw image. However Adobe's Digital Negative (DNG) format [83] appears to be quite popular [82, 84].

2.4 Conclusion

In this chapter, we discussed colour image formation model, human visual system (HVS) and digital imaging system. We explored how these components are linked together. In the discussion on the image formation model, we described how the interaction between illuminant, surface and sensor (both HVS and imaging sensor) is mathematically modelled. In the discussion on HVS, we gave a brief description on the structure of HVS. We then discussed how the responses from the HVS was first measured, and how it is represented mathematically. In the discussion on digital imaging system, we gave a brief description on the image processing pipeline of a digital camera. We discussed the steps involved in creating a digital image. The process that convert the raw colour response from a digital imaging system and the HVS-linked mathematical representation is colour correction.

There are two factors which affects the results of colour correction. The process of colour correction can be affected by the algorithm used, as well as the dataset which is used to train the colour correction algorithm. In the next chapter, Chapter 3, we will have a review of common colour correction algorithms. In Chapter 4, we will look at the reflectance datasets which are used to train and evaluate colour correction algorithms.

Chapter 3

Evaluation of commonly used colour correction algorithms

3.1 Introduction

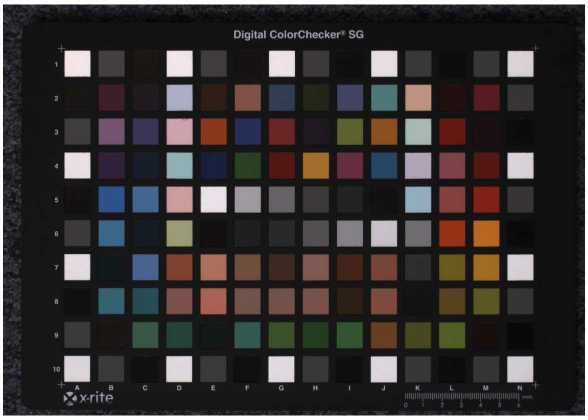
This chapter is an extended version of a published conference paper [85].

The problem of colour correction arises from the fact that cameras do not measure colour in the same way as the Human Visual System (HVS). The imaging sensor in a camera does not have the same spectral sensitivity as the cone cells in the human eyes. The problem is aggravated by the fact that different models of cameras also use different imaging sensors. Therefore, to reproduce colour responses as perceived by the HVS, it is necessary to convert device specific responses produced by a particular camera to a device independent colour space linked to the HVS.

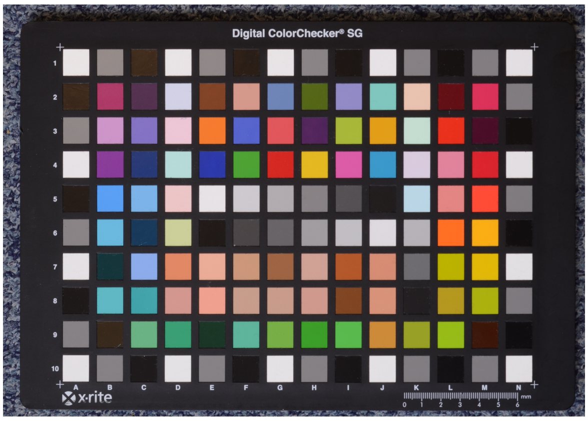
Conceptually colour correction in a digital camera is a two step process [14]. The colour responses are first transformed from the device specific colour space to an unrendered colour space, which is typically the CIE XYZ (introduced in subsection 2.2.3). Then, the obtained values are converted from the chosen unrendered colour space to a rendered colour space, which is typically the sRGB [86]. The conversion from device specific colour space to CIE XYZ is never a perfect process, as the device specific colour space is typically not a linear combination of CIE XYZ. We will discuss this further later in the chapter.

Typically, the colour correction procedure involves measuring the device response for some physical targets, often viewed with respect to multiple lights, then perform a correction towards the actual colour values of the same targets on the device-independent colour space. For example, Figure 3.1a shows a raw image of a colour checker captured by a Nikon D5100 camera, and Figure 3.1b shows the same image after it is corrected to sRGB colour space.

It is clear that the colours recorded by the camera are significantly different after colour correction.



(a) Raw image from Nikon D5100



(b) Same image after being corrected to sRGB

Fig. 3.1 Nikon D5100 raw camera response of an image containing a colour checker [87], before (a) and after (b) correction to sRGB colour space by means of a 3×3 colour correction matrix. Both images have a gamma of 0.45 applied (standard for the sRGB case). Note the washed out colour in (a).

The sRGB colour space is constructed as a linear combination of the CIE XYZ colour matching function. Similar to XYZ, sRGB is a linear space – doubling the light intensity doubles the sRGB responses. To make a reproduction the sRGB, a fractional gamma term is applied.

The XYZ colour space is widely used in colour sciences when the aim is to automatically grade reproduction performance. The XYZ coordinate system underpins commonly used perceptually colour spaces including CIELAB and CIELUV [33]. Throughout this thesis we map camera colours to XYZ and judge colour correction performance relative to this space.

Colour correction is never perfect. In order to achieve perfect colour correction, the spectral sensitivities of the camera must be a linear transform from the CIE XYZ colour matching functions, in other words the Luther conditions must hold [88]. As an example, Figure 3.2a shows the CIE XYZ standard human observer colour matching function, while Figure 3.2b shows the spectral sensitivities of an Nikon 5100 digital camera [89]. From these two plots, it is clear that the spectral sensitivities from the camera are far away from those of XYZ. Figure 3.2c shows the result of applying the best linear least square transform to the camera spectral sensitivities in order to match the CIE XYZ curves. However, we can clearly see that the curves in Figure 3.2c are still different than those in Figure 3.2a. This shows that colour correction is not perfect, and thus that The Luther condition does not hold.

Different colour correction algorithms have different advantages and disadvantages, as they were designed with different priorities in mind. Before we look into different algorithms individually, we need to first look at how to capture the data used in colour correction algorithm training.

There are two approaches for training colour correction algorithms: the *empirical* and the *synthetic* approach. In the empirical approach, we regress observed camera RGBs to previously measured target XYZs. Typically the training target is the colour checkers [87] (as shown in Figure 3.1). This approach presents one main drawback, to achieve the best result, each patch on the colour checker needs to receive the same amount of illumination. There are two techniques to solve this problem. First, we could try and make the illumination uniform. However uniform light is hard to achieve in real-life condition as well as inside a laboratory.

The second technique, aims at measuring then discounting the variation of illumination on the colour checker. A grey chart at the same location of the colour checker, and capturing an image of it. Then the variation of illumination on the colour checker can be removed by dividing the image of the colour checker by the image of the grey chart. However, it can be difficult to place the grey chart at the exact same location as the colour checker. The empirical approach also has another problem – the reflectance of the patches on the colour checker may change over time. Furthermore,the grey chart and the colour checker may develop warping over time, due to the temperature and humidity of the environment. All of these issues make capturing data to train colour correction algorithm empirically a rather laborious and delicate task.

In contrast, the *synthetic* approach for colour correction is based on the camera's spectral sensitivity, measured reflectance spectra, measured illuminant spectra and the mathematical model of image formation. Rather than measuring the camera's RGB response by taking pictures, they are calculated using the image formation model. The image formation model

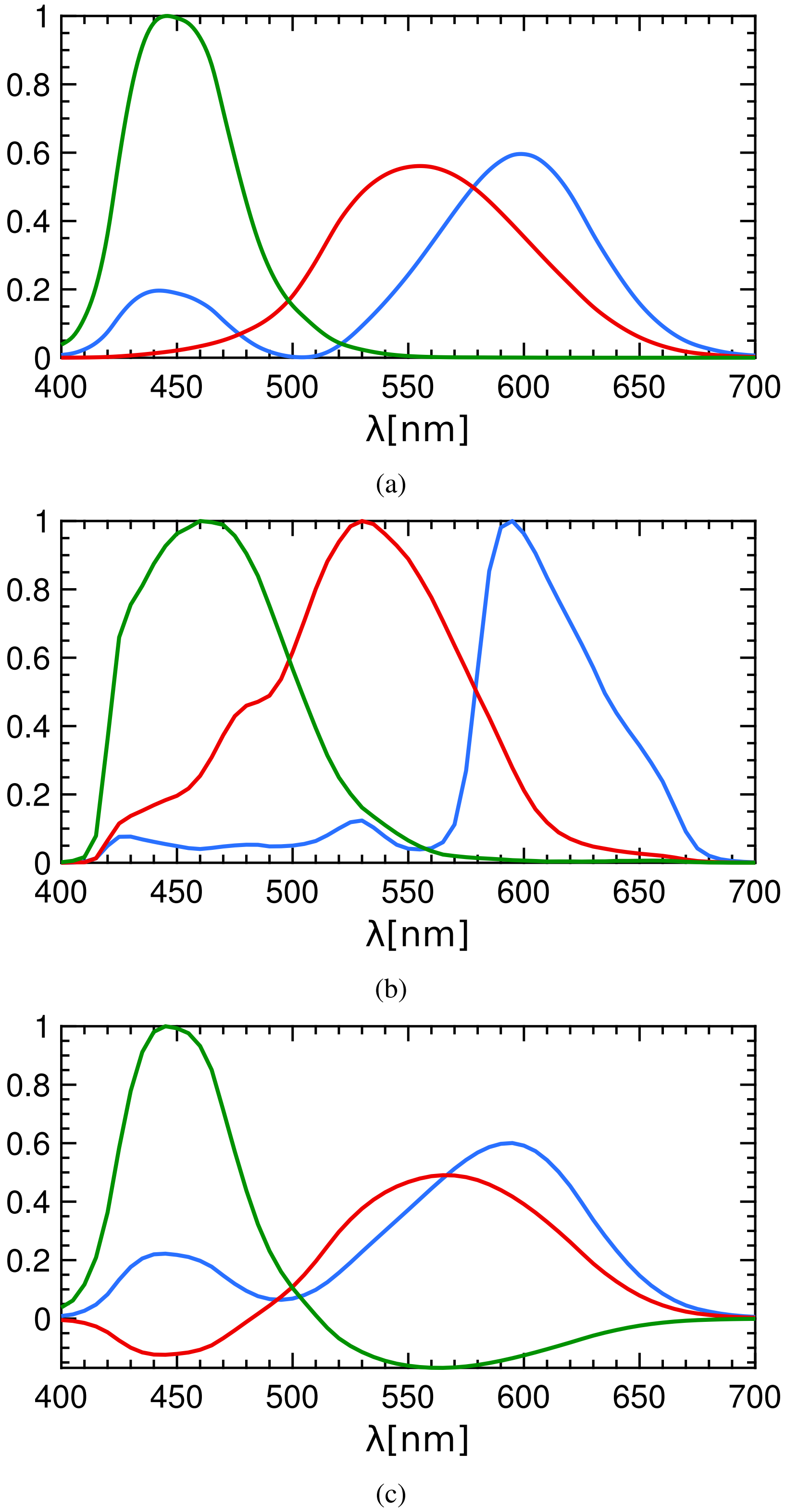


Fig. 3.2 (a) The CIE XYZ colour matching function; (b) The spectral sensitivities for Nikon D5100) [89]; and (c) CIE XYZ estimated from the spectral sensitivities of Nikon D5100 using linear least square transform.

was reviewed in section 3.3. Advantageously, given measured data it is easy to create ‘images’ of many samples. This means that the problems associated with taking physical images (e.g. varying illumination) no longer applies. Conversely, measurement of spectra takes time and recovering camera device sensitivities is a laborious process. Of course, in terms of the measured spectra we can use one of the many set of reflectances and illuminants that are publicly available. However, the question of which spectral dataset should be used remains. This is a question we address in some detail in this thesis.

3.2 The experimental dataset used in this thesis

In this thesis, we mainly use the synthetic approach for colour correction. This allows us to generate a large amount of device specific RGBs under different illuminants conditions, and their corresponding XYZs. To do so, we extensively use previously published datasets.

Regarding the reflectance datasets, we consider:

- Agfa ColorReference IT8.7/2 [90] – This dataset contains 288 reflectance measurements in total. It is designed according the ANSI IT8.7/2 standard [91], which describes the layout and colorimetric values of a input calibration target intended for a photographic paper / scanner combination. The standard is designed in such a way so the calibration target can be manufactured using a colour photographic paper with photographic medium dye.
- Natural Dataset [92] – This dataset measured by Westland *et al* comprises 404 measured spectra of plants, foliage and flowers.
- 1269 Munsell colour chips [93] – This dataset contains the reflectance measurements from 1269 colour chips from Munsell Book of Colour, Matte Edition [94]. Munsell Book of Colour covers a wide range of colour, this leads to its wide-spread usage.
- Macbeth ColorChecker [95] – This dataset contains 24 reflectance measurements. Macbeth ColorChecker was originally developed to facilitate quantitative and visual evaluation processes employed in photography, television and printing.
- ColorChecker Digital SG [87] – This dataset contains 96 reflectance measurements. ColorChecker Digital SG is an extension to the original Macbeth ColorChecker. It is a commonly used calibration target for digital cameras and scanners.
- 120 DuPont paint chips [96] – the chips were selected from the Solid Color Selection of the DuPoint Color Sampler. This dataset contains commonly used artificial colour.

- 170 Objects [96] – the selected sample include various natural and man-made objects, including rocks, plants and vegetation, human skin and hair, and fabrics.

We use the following camera spectral sensitivities dataset:

- A camera spectral sensitivity dataset consists of 28 cameras [88]
- Sony DXC-930 three chip CCD video camera [97]
- Nikon D5100 DSLR [89]
- Sigma SD1 Merill DSLR [89]

Finally we consider the following illuminant datasets:

- Illuminant series D [98], which represents various phases of daylight between 4800 K and 10000 K. In particular, the CIE Standard Illuminant D65 is intended to represent average daylight and has a correlated colour temperature of approximately 6500 K. It is recommended to be used in all colorimetric calculations requiring representative daylight [35].
- Illuminant A [99], which is intended to represent typical, domestic, tungsten-filament lighting. Its relative spectral power distribution is that of a Planckian radiator at a temperature of approximately 2856 K. CIE standard illuminant A is recommended for all applications of colorimetry involving the use of incandescent lighting.
- SFU illuminant database [97], which consists of the spectra of 11 various light sources and 82 spectra measured in and around the Simon Fraser University campus at various times of the day, and in a variety of weather conditions.

3.3 Colour formation

Unless stated otherwise, we adopt the following convention in this thesis regarding image formation:

- All spectra (including illuminant, reflectance, colour matching function and camera spectral sensitivities, etc.) are sampled between 400 nm and 700 nm at a 10 nm interval. This means that we obtain 31 discrete samples for each spectrum. For the spectra that were originally not sampled at the wavelength and interval stated above, we perform resampling using monotone piecewise cubic interpolation [100].

- **S** is a $31 \times n$ matrix representing a set of n surface reflectance spectra. Each row of the matrix represents the reflectivity at a certain wavelength, and each column represents a different surface sample.
- **X** and **R** are 31×3 matrices containing the CIE XYZ colour matching function and the device spectral sensitivities respectively;
- **E** is an $n \times n$ diagonal matrix, where each entry in the diagonal represents the intensity of the illuminant at a particular wavelength.
- **P** and **Q** are $3 \times n$ matrices representing the camera responses and the CIE XYZ tristimulus values to the entire calibration dataset.

The camera and the CIE XYZ responses to the illuminant **E** and reflectances **S** are captured by the $3 \times n$ matrices **P** and **Q**, that can be computed by:

$$\mathbf{P} = \mathbf{R}^{\mathsf{T}}(\mathbf{ES}) \tag{3.1a}$$

$$\mathbf{Q} = \mathbf{X}^{\mathsf{T}}(\mathbf{ES}). \tag{3.1b}$$

The problem of colour correction can be considered as finding ways to map the values of **P** to those of **Q**, i.e. finding a function $f(\cdot)$ such that $f(\mathbf{P}) \approx \mathbf{Q}$.

3.4 Evaluation framework for colour correction algorithms

We focus on the synthetic approach to evaluate the performance of colour correction algorithms. In particular, we proceed as follows:

1. For a chosen illuminant (e.g. D65), we compute the CIE XYZ tristimulus values for the chosen reflectance datasets.
2. The corresponding device specific RGBs for the chosen cameras is then computed.
3. We use three-fold cross-validation to select the training set and the testing set. Three-fold cross validation was chosen because it is quite common in colour research [101–103].
4. The chosen colour correction algorithm is trained using the training dataset.
5. We then apply the colour correction algorithm to the test dataset, converting device specific RGBs to estimated XYZs.

6. Both the estimated XYZs and the true XYZs are converted into CIELAB colour space. The CIELAB ΔE^*_{ab} colour distance metric is then computed.

3.5 Colour correction algorithms

There are many methods for relating the device specific responses to CIE XYZ. In this chapter, we look at the following colour correction methods:

- Linear least squares
- Polynomial regression
- Root polynomial regression
- Homography
- Angular minimisation
- Hue plane preserving
- Maximum ignorance
- Maximum ignorance with positivity

There are other colour correction algorithms, e.g. look-up tables [104] or neural networks [105], but they are outside the scope of this thesis.

3.5.1 Linear Least Square Colour Correction

Linear least squares colour correction (LLSCC) is the simplest and most commonly used method for colour correction. The aim of LLSCC is to find the best 3×3 matrix $\mathbf{M}$ such as the relation between the device specific RGBs and the XYZ tristimulus values can be written as:

$$\mathbf{Q} = \mathbf{M}\mathbf{P}\,. \tag{3.2}$$

Therefore, we look for the 3×3 matrix M that minimises:

$$\min_{\mathbf{M}}\{||\mathbf{M}\mathbf{P} - \mathbf{Q}||_F\}\;, \tag{3.3}$$

where $||\cdot||_F$ above denotes Frobenius norm.

The most common solution for this problem – the ordinary least-squares 3 × 3 regression states that matrix **M** can be solved by the Moore-Penrose pseudoinverse:

$$\mathbf{M} = \mathbf{Q}\mathbf{P}^{\mathsf{T}}(\mathbf{P}\mathbf{P}^{\mathsf{T}})^{-1}. \tag{3.4}$$

The matrix **M** can be thus understood as the one that best maps the colour values of a particular dataset in the ordinary least-square sense. The idea of using linear regression to establish a mapping between RGB and XYZ was first introduced by Horn [10].

The popularity of LLSCC is partly due to the fact that it is exposure invariant. Exposure invariance is a desirable property for a colour correction algorithm that is used in a digital camera. This is because in most practical applications involving digital cameras, the length of exposure is uncontrollable and unpredictable, and they are expected to operate across different lighting conditions.

Let us mathematically explain the exposure invariant property of LLSCC. Suppose that matrix **M** maps an RGB $\mathbf{p}$ to its corresponding XYZ $\mathbf{q}$. Now, we double the intensity of the light yielding a linear raw RGB response of $2\mathbf{p}$. The correct colour correction here is still **M** since $\mathbf{M}(2\mathbf{p}) = 2\mathbf{q}$. With some exceptions (e.g. [106, 107]), most non-linear colour corrections are not exposure invariant.

Another factor for the popularity of LLSCC is that linear colour correction is easy to implement. After training the colour correction matrix, the colour correction step itself is a simple 3 × 3 matrix multiplication. Moreover, the conversion from XYZ to the sRGB colour space (the output space of most cameras) is modelled as a 3 × 3 matrix multiplication plus a non-linear gamma correction. For this reason, quite often, the colour correction matrix and the XYZ-to-sRGB conversion matrix are multiplied together to form a single 3 × 3 matrix, in order to reduce the number of computational steps required. In summary, LLSCC provides reasonable performance under most conditions.

3.5.2 Polynomial Colour Correction

In order to reduce the colour reproduction error obtained by LLSCC, one can use polynomial colour correction (PCC). In PCC, polynomial regression is used to find the colour correction matrix, rather than ordinary least square regression. Practically, this is achieved by adding extra rows containing polynomial terms in the minimisation equation shown for LLSCC (**P** of Equation 3.2). By doing this, we obtain the following equation:

$$\min_{\mathbf{M}}\{\|\tilde{\mathbf{M}}\tilde{\mathbf{P}} - \mathbf{Q}\|_F\}\ , \tag{3.5}$$

where $\tilde{\mathbf{M}}$ is the polynomial colour correction matrix, $\tilde{\mathbf{P}}$ is the device response matrix with rows of polynomial terms added.

The explicit solution for Equation 3.5 using the Moore-Penrose pseudoinverse is:

$$\mathbf{M} = \mathbf{Q}\mathbf{P}^{\intercal}(\mathbf{P}\mathbf{P}^{\intercal})^{-1}. \tag{3.6}$$

Let us denote the polynomial terms using $\boldsymbol{\rho}_{k,n}$, where k is the polynomial order, and n is the number of channels on the imaging sensor. The 2nd, 3rd and 4th polynomial terms of an imaging sensor with 3 channels are:

$$\begin{aligned}
&\boldsymbol{\rho}_{2,3} : (r, g, b, r^2, g^2, b^2, rg, rb, gb)^{\intercal} \\
&\boldsymbol{\rho}_{3,3} : (r, g, b, r^2, g^2, b^2, rg, rb, gb \\
&\qquad r^3, r^2g, r^2b, rg^2, rgb, rb^2, g^3, g^2b, gb^2, b^3)^{\intercal} \\
&\boldsymbol{\rho}_{4,3} : (r, g, b, r^2, g^2, b^2, rg, rb, \\
&\qquad r^3, r^2g, r^2b, rg^2, rgb, rb^2, g^3, g^2b, gb^2, b^3, \\
&\qquad b^4, gb^3, g^2b^2, g^3b, g^4, rb^3, rgb^2, rg^2b, rg^3, r^2b^2, r^2gb, r^2g^2, r^3b, r^3g, r^4)^{\intercal}
\end{aligned} \tag{3.7}$$

Therefore matrices $\tilde{\mathbf{M}}$ and $\tilde{\mathbf{P}}$ respectively have dimension $3 \times t$ and $t \times n$, where $t = 9, 19, 37$ for orders 2, 3, and 4.

Formally, the setup of K^{th} degree polynomial terms in N variables is defined as:

$$\boldsymbol{\rho}_{k,n} = \{\prod_{i=1}^{n} \rho^{\alpha_i : \Sigma a_i \leq k}\}, \tag{3.8}$$

It can be shown that the number of polynomial terms up to k^{th} degree is $\binom{n+k-1}{k}$.

Polynomial terms beyond 4th degree can be enumerated, however they are not often used. This is because the issue of overfitting. Overfitting happens when the resulting model contains more parameters than those that are justifiable by the training data.

An overfitted colour correction matrix has a really low CIELAB ΔE error when applied to the training dataset. However, the error is very large when applied to a testing dataset. Additionally, when an overfitted colour correction matrix is applied to a RAW image, the noise from the raw image is amplified, which results in unpleasant artefacts in the colour corrected image.

Overfitting can be avoided by using regularisation techniques. A commonly used regularisation technique is the Tikhonov regularisation. Here, a regularisation term $||\boldsymbol{\Gamma}\tilde{\mathbf{M}}||_F$ is incorporated into the minimisation equation(Equation 3.5). In this way we obtain:

$$\min_{\mathbf{M}}\{||\tilde{\mathbf{M}}\tilde{\mathbf{P}}-\mathbf{Q}||_F+||\mathbf{\Gamma}\tilde{\mathbf{M}}||_F\}\ , \tag{3.9}$$

where $\mathbf{\Gamma}$ is denoted as the Tikhonov matrix.

Tikhonov matrix is typically chosen as a multiple of the identity matrix, i.e. $\mathbf{\Gamma}=\alpha\mathbf{I}$, where α is a positive scalar (typically a small number). As we increase α the magnitude of the fitting matrix is significant if it has large values. Put another way, the penalty term encourages the fitting matrix to have bounded norm. Bounded-norm fits are more likely to generalise to unseen data.

The optimal α can be chosen by search (where the user seeks to trade off the fitting error and the magnitude of the recovered fitting matrix). Tikhonov regularisation is also known as L_2 regularisation.

The explicit solution for Equation 3.9 using Moore-Penrose pseudoinverse is:

$$\tilde{\mathbf{M}}=\mathbf{Q}\tilde{\mathbf{P}}^{\intercal}(\tilde{\mathbf{P}}\tilde{\mathbf{P}}^{\intercal}+\mathbf{\Gamma}\mathbf{\Gamma}^{\intercal})^{-1}. \tag{3.10}$$

PCC is not exposure invariant. It is therefore not practical to use it in digital cameras for photography. This is because digital cameras are expected to operate in environment with varied brightness. PCC can be used in conditions where imaging conditions can be controlled, e.g. document and photo scanners.

3.5.3 Root-Polynomial Colour Correction

Root polynomial colour correction (RPCC) [106] was proposed to provide better performance than LLSCC, while at the same time preserving the important property of exposure invariance. This is achieved by adding rows with root-polynomial terms into $\mathbf{P}$ in Equation 3.2, instead of polynomial terms.

Let us denote the root-polynomial terms using $\bar{\boldsymbol{\rho}}_{k,n}$, where k is the order of the root-polynomial, and n is the number of channels on the imaging sensor. The 2nd, 3rd and 4th root-polynomial terms of an imaging sensor with 3 channels are:

$$\begin{aligned}
&\bar{\boldsymbol{\rho}}_{2,3} : (r, g, b, \sqrt{rg}, \sqrt{gb}, \sqrt{rb})^{\intercal} \\
&\bar{\boldsymbol{\rho}}_{3,3} : (r, g, b, \sqrt{rg}, \sqrt{gb}, \sqrt{rb}, \\
&\qquad \sqrt[3]{rg^2}, \sqrt[3]{gb^2}, \sqrt[3]{rb^2}, \sqrt[3]{gr^2}, \sqrt[3]{bg^2}, \sqrt[3]{br^2}, \sqrt[3]{rgb})^{\intercal} \\
&\bar{\boldsymbol{\rho}}_{4,3} : (r, g, b, \sqrt{rg}, \sqrt{gb}, \sqrt{rb}, \\
&\qquad \sqrt[3]{rg^2}, \sqrt[3]{gb^2}, \sqrt[3]{rb^2}, \sqrt[3]{gr^2}, \sqrt[3]{bg^2}, \sqrt[3]{br^2}, \sqrt[3]{rgb} \\
&\qquad \sqrt[4]{r^3g}, \sqrt[4]{r^3b}, \sqrt[4]{g^3r}, \sqrt[4]{g^3b}, \sqrt[4]{b^3r}, \sqrt[4]{b^3g}, \\
&\qquad \sqrt[4]{r^2gb}, \sqrt[4]{g^2rb}, \sqrt[4]{b^2rg})^{\intercal}
\end{aligned} \tag{3.11}$$

Similar to PCC, the higher order root-polynomial terms can be derived, although they are not used often due to the the overfitting problem. However, RPCC is less susceptible to overfitting compared to PCC. This is because for each root-polynomial order, there are fewer parameters.

In terms of CIELAB error, RPCC performs better than LLSCC, but worse than PCC. However it should be noted that there are fewer variable terms in RPCC compared to PCC, i.e. the number of root-polynomial terms in RPCC is fewer than the number of polynomial terms in PCC. RPCC tends to produce less noisy images as it is less susceptible to overfitting. The most important advantage of RPCC over PCC is that RPCC is exposure invariant, similar to LLSCC. This provides the advantage that a brightness change in the imaging environment does not result in hue shifts for the colour corrected image. This means that RPCC is suitable for used in digital cameras.

3.5.4 Hue Plane Preserving Colour Correction

A hue plane is a geometrical half-plane defined by the neutral axis and a chromatic colour. In this way, hue planes can be used for separating the colour space into different regions, each of them having similar colour properties. In particular, in hue plane preserving colour correction, the colour spaces are divided in sub-regions delimited by two hue planes. In order to map RGBs to XYZs, a 3×3 matrix is learned and applied in each subregion separately [107]. These matrices can also be constrained to preserve the whitepoint. The sub-regions can also be flexibly chosen in number and position to regularise and optimise results, while constraining continuity across hue planes. Hue plane preserving colour correction provide significantly higher colorimetric accuracy compared to linear colour correction, while maintaining exposure invariance. Its performance is comparable to root-polynomial colour correction. This disadvantage is that this algorithm is formulated in a mathematically more complex way.

3.5.5 Colour Correction by Angular Minimisation

As discussed in section 3.1, most colour correction algorithms are sensitive to the brightness difference within the training RGBs and XYZs. In Colour Correction by Angular Minimisation (CCAM), RGB and XYZ triplets are treated as two sets of vectors. The angular differences between these two sets of vectors are minimised [108]. The magnitude of these two sets of vectors are ignored during the optimisation. This means that colour correction by angular minimisation is resistant to brightness difference within the training dataset.

CCAM finds a 3×3 linear transform $\mathbf{M}$ which map RGB to XYZ. Considering a set of device specific RGBs $\{\mathbf{p}\}_{i=1}^{N}$, and their corresponding XYZ tristimulus values $\{\mathbf{q}\}_{i=1}^{N}$, the formulation for the minimisation is the following:

$$\sum_{i=1}^{N} \cos^{-1}\left(\frac{\mathbf{M}\mathbf{p}_i \cdot \mathbf{q}_i}{|\mathbf{M}\mathbf{p}_i||\mathbf{q}_i|}\right) , \tag{3.12}$$

where ‘ $\cdot$ ’ denotes the vector dot-product. This algorithm is based on the idea of dot-product vectors – the dot-product of two vectors divide by the product of their magnitudes equals to the cosine of the angle between them. Since the vector magnitudes are ignored in the minimisation, the magnitude of $\mathbf{M}$ can be arbitrarily small or large. In order to not alter the overage image brightness, $\mathbf{M}$ is rescaled so that the sum of all its elements equals 3 in sRGB space.

3.5.6 Homography Colour Correction

In mathematics, a homography is a mapping between two projective spaces. In computer vision, homography is typically used in tasks such as image registration and stereo vision. This is because if there are two images of the same object taken at different perspectives and/or with different cameras, they are related to each other via a homography transform. In homography colour correction (HMCC) [109], the transformation between RGB and XYZ is treated as a problem involving the mapping of chromaticities in homogeneous coordinates, which is then solved using homography.

As in the previous cases, HMCC assumes Lambertian image formation. The relationship between device specific RGB and XYZ tristimulus values is written as the following:

$$\mathbf{DPH} \approx \mathbf{Q} , \tag{3.13}$$

where **D** is an $n \times n$ diagonal matrix of scalar shading factor, and **H** is a 3×3 homography colour correction matrix. If both **D** and **H** are applied to **P**, then this is called *shading homography*.

Equation 3.13 is solved using alternating least squares algorithm [110, 109]. This algorithm iterates between optimising for shading difference and for chromaticity difference for a final output. The colour correction matrix generated by homography colour correction is exposure invariant. A uniform lighting field is also not required during the capture of training data as the variation is captured by the matrix **D**. This method discards less information during the training phase, compared to colour correction by angular minimisation.

3.5.7 Maximum Ignorance Colour Correction

The maximum ignorance colour correction (MICC) is an approach which operates without an explicit calibration data set. Instead, the transform used for colour correction is defined to be the mapping which best takes the device response functions onto the XYZ matching curves. The statistical assumption made here is that all possible spectra, with both positive and negative power at each wavelength, all occur with equal likelihood. This approach can be justified, as [10, 111] have shown, in that perfect colour correction for any colour stimulus is possible if and only if the device sensitivities are a linear transform from the colour matching functions. The maximum ignorance assumption is useful in practice, as the CIELAB ΔE^*_{ab} for this method can be as low as 4. However, the statistical assumption made here is of course flawed, since spectra with negative power cannot physically exist.

In section 3.3, we presented the equation for the formation of device specific RGB responses, and CIE XYZ tristimulus responses. By defining the colour signal as $\mathbf{C} = \mathbf{ES}$, we have:

$$\mathbf{P} = \mathbf{R}^{\mathsf{T}}\mathbf{C} \tag{3.14a}$$

$$\mathbf{Q} = \mathbf{X}^{\mathsf{T}}\mathbf{C} \, . \tag{3.14b}$$

We showed that linear least square colour correction uses a 3×3 matrix **M** to transform device specific RGB to CIE XYZ:

$$\mathbf{Q} = \mathbf{MP} \, . \tag{3.15}$$

We also showed the ordinary least square solution for this problem is:

$$\mathbf{M} = \mathbf{QP}^{\mathsf{T}}(\mathbf{PP}^{\mathsf{T}})^{-1} \, . \tag{3.16}$$

By substituting Equation 3.14 into Equation 3.16, we obtain:

$$\mathbf{M} = \mathbf{X}^{\mathsf{T}}\mathbf{C}\mathbf{C}^{\mathsf{T}}\mathbf{R}(\mathbf{R}^{\mathsf{T}}\mathbf{C}\mathbf{C}^{\mathsf{T}}\mathbf{R})^{-1} \,. \tag{3.17}$$

This shows that the knowledge of the colour signal is required for calculating the camera sensor response, which is needed for calculating the colour correction matrix.

The aim of MICC is to compute the best linear correction matrix, when the information about the colour signal is unavailable. We assume no knowledge of the surface reflectance of the scene, or the spectral curve of the illuminant. The only required information is the sensor response curve for the imaging device. We make the assumption that the colour signal is a random variable following uniform distribution with values between $[-1,1]$. This means that the vector **c** in Equation 2.6 is filled with uniformly distributed random values drawn between $[-1,1]$ [112].

Rather than computing the $\mathbf{C}\mathbf{C}^{\mathsf{T}}$ multiplication using data, we can compute the *expectation* of $\mathbf{C}\mathbf{C}^{\mathsf{T}}$, denoted as $E(\mathbf{C}\mathbf{C}^{\mathsf{T}})$. Matrix $\mathbf{C}\mathbf{C}^{\mathsf{T}}$ contains the variance and covariance for the colour signal readings at difference frequencies. It has the following structure:

$$\begin{bmatrix} E(c_1^2) & E(c_1c_2) & \dots & E(c_1c_{31}) \\ E(c_2c_1) & E(c_2^2) & \dots & E(c_2c_{31}) \\ \vdots & \vdots & \ddots & \\ E(c_{31}c_1) & E(c_{31}c_2) & \dots & E(c_{31}^2) \end{bmatrix}, \tag{3.18}$$

where c_i is the i-th row of **C**. The major-diagonal entries of Equation 3.18 contain the variance of c_i. We can calculate their numerical values by using the following approach: let a be a continuous random variable, the variance of that variable can be calculated using integration:

$$\begin{aligned} &\int_{-1}^{1} a \cdot a \, da \\ &\int_{-1}^{1} a^2 \, da \\ &= \left[\frac{a^3}{3}\right]_{-1}^{1} \\ &= \frac{2}{3} \end{aligned} \tag{3.19}$$

The off-diagonal entries of Equation 3.18 contains the covariance between c_i and c_j. This covariance can also be computed by integration: let a and b be two unrelated continuous random variables representing the colour signal, the covariance between a and b can be calculated as:

$$\begin{aligned}
&\int_{-1}^{1}\int_{-1}^{1} ab\;da\;db \\
&= \left[\left[\frac{a^2b^2}{4}\right]_{-1}^{1}\right]_{-1}^{1} \\
&= \left[\frac{b^2}{4}\right]_{-1}^{1} \\
&= 0
\end{aligned} \tag{3.20}$$

By substituting the results from Equation 3.19 and Equation 3.20 into Equation 3.18, we obtain a 31 × 31 identity matrix multiplied by $\frac{2}{3}$. By ignoring the scaling factor, we can replace the raw cross product matrix $\mathbf{CC}^\intercal$ in Equation 3.17 with a 31 × 31 identity matrix. Equation 3.17 becomes:

$$\mathbf{M} = \mathbf{X}^\intercal\mathbf{R}(\mathbf{R}^\intercal\mathbf{R})^{-1}. \tag{3.21}$$

Therefore, the problem of maximum ignorance colour correction effectively becomes linear regression problem between the camera's spectral sensitivity curve and the CIE XYZ colour matching function.

3.5.8 Maximum Ignorance with Positivity Colour Correction

Maximum ignorance with positivity colour correction (MIPCC) is similar to MICC, in the sense that it does not require an explicit calibration data set. The major difference for this method is that the colour signal is assumed to be strictly positive and equally likely [112]. MIPCC improves on the conventional MICC zero-calibration method by providing a better statistical assumption, as negative spectral power does not make physical sense. Practically MIPCC provides a substantially improved colour correction performance.

MIPCC is achieved by computing a different expected colour signal autocorrelation matrix ($E(\mathbf{CC}^\intercal)$). We make the assumption that the colour signal is a random variable following uniform distribution with values between $[0,1]$.

The major-diagonal entries of $E(\mathbf{CC}^\intercal)$ contain the variance of c_i, and can be computed as follows: Let a be a continuous random variable, the variance of variable a can be calculated using integration:

$$\begin{aligned} &\int_0^1 a \cdot a \, da \\ &\int_0^1 a^2 \, da \\ &= \left[\frac{a^3}{3}\right]_0^1 \\ &= \frac{1}{3} \end{aligned} \tag{3.22}$$

Also as before, the off-diagonal entries $E(\mathbf{CC}^\mathsf{T})$ contains the covariance between c_i and c_j and can be compued as follows: Let a and b be two unrelated continuous random variables, the covariance between a and b can be calculated using integration:

$$\begin{aligned} &\int_0^1 \int_0^1 ab \, da \, db \\ &= \left[\left[\frac{a^2 b^2}{4}\right]_0^1\right]_0^1 \\ &= \left[\frac{b^2}{4}\right]_0^1 \\ &= \frac{1}{4} \end{aligned} \tag{3.23}$$

The resulting $E(\mathbf{CC}^\mathsf{T})$ has the following structure:

$$\begin{bmatrix} \frac{1}{3} & \frac{1}{4} & \cdots & \frac{1}{4} \\ \frac{1}{4} & \frac{1}{3} & \cdots & \frac{1}{4} \\ \vdots & \vdots & \ddots & \\ \frac{1}{4} & \frac{1}{4} & \cdots & \frac{1}{3} \end{bmatrix}, \tag{3.24}$$

We will revisit the idea of finding the expectation of an autocorrelation in Chapter 5.

3.6 Experiment

In this chapter, we performed a colour correction experiment using spectral data and camera responses measured ourselves, to demonstrate the performance of the colour correction algorithms described. We took reflectance spectra measurements from a SG140 colour checker, under a D65 illuminant, using a Photo Research PR-670 spectroradiometer. The colour checker was then photographed using a Nikon D5100 camera. In this particular colour correction experiment, we decided to use empirically measured camera responses rather than synthetically generate camera responses. This provides us with real-world performance metrics, which acts as a sanity check for the later chapter, in which we use synthetically generated camera responses for studying colour correction algorithms. The colour correction experiments itself was performed using three-fold cross-validation. The results of the experiment are shown in Table 3.1.

Table 3.1 CIELAB ΔE^*_{ab} for Colour Correction Algorithms Described in this Chapter

Method	**Mean**	**Median**	**95%**	**Max**
Linear least squares	2.93	2.09	8.31	26.99
Second order polynomial	2.34	1.71	7.10	10.36
Third order polynomial	**1.77**	1.43	**4.07**	6.30
Second order root-polynomial	2.05	1.72	4.45	6.42
Third order root-polynomial	2.66	**1.34**	6.65	**6.18**
Homography	2.71	2.23	6.15	15.82
Hue plane preserving	2.09	1.61	5.29	13.44
Angular minimisation	2.80	2.07	6.66	15.74
Maximum ignorance	5.32	4.46	12.22	13.61
Maximum ignorance with positivity	3.99	3.66	8.25	9.35

These results show that every method other than the maximum-ignorance based approaches outperforms linear least square colour correction. This is because linear least square colour correction is the simplest, and every other algorithms were designed to improve upon linear least square colour correction.

The algorithm with the best performance is the third order polynomial colour correction. The second order root-polynomial colour correction performs better than second order polynomial colour correction, while the third order polynomial colour correction performs better that the third order root-polynomial colour correction for the mean and the 95% measures. Therefore, root-polynomial colour correction provides comparable performance compared to polynomial colour correction. However, it should be noted again that root-polynomial based approaches are exposure invariant, which means that they are not affected by changes in the scene brightness. This makes root-polynomial based approaches more versatile than polynomial based approaches. The numerical value of the benchmark does not tell the full story.

Homography colour correction performs slightly better than colour correction by angular minimisation, in terms of mean CIELAB ΔE^*_{ab} . They both perform slightly better than linear least square colour correction. They both have the advantage that they do not require an image of the grey chart, or the training colour checker to be uniformly illuminated. These two conditions can be hard to achieve in practice.

Hue-plane preserving colour correction produces error comparable to second order root-polynomial colour correction. However it requires a lot of parameters, as the colour space is divided based on hue planes, and a linear least square colour correction matrix is produced for each hue plane.

Maximum ignorance-based colour correction methods produce higher CIELAB ΔE^*_{ab} than other methods. This is because they assume no prior information about the colour signal. However, their performance is more consistent across datasets.

3.7 Conclusion

In this chapter, we looked at how experimental data for colour correction can be captured, and we briefly introduced the experimental data used in this thesis. We then discussed a few commonly used colour correction algorithms. In the next chapter, we will perform colour correction experiments using different reflectance datasets, and study their differences.

Chapter 4

Measuring the differences between reflectance datasets

4.1 Introduction

The performance of colour correction of a digital imaging device depends on two factors – the algorithm used and the dataset used for training the algorithm. In the previous chapter (Chapter 3), we introduced the datasets used in this thesis, we then reviewed various colour correction algorithms. We performed colour correction experiment using spectral data and camera responses measured ourselves. In this chapter, we look at the various reflectance data introduced by other authors.

Reflectance datasets are heavily used in colour science research. In the context of colour correction, reflectance datasets are used to train colour correction algorithms, typically resulting in colour correction matrices. The training reflectance dataset does have an impact on the resulting colour correction matrix. For this reason, it is widely accepted that the performance of a colour correction matrix is best when the reflectance of the scene matches with the training dataset, but it may suffer more errors when this is not the case.

Perhaps the most commonly used reflectance dataset is the combined SFU reflectance dataset, which contains 1995 reflectance spectra by Barnard et al. [97]. The SFU reflectance dataset itself is a combination of multiple reflectance datasets, including 1269 Munsell colour chips [94], 120 Dupont paint chips [96], 170 natural objects [96], 350 reflectances from natural formations [113], and 57 additional surfaces measured by Barnard et al. themselves. However, the composition of this reflectance dataset is fairly arbitrary as Barnard et al. did not provide a justification for the choice of the reflectance dataset they included.

In this chapter, we investigate the differences between various reflectance datasets and how these differences affect colour correction. In more detail, we attempt to answer the following question: Given two reflectance datasets, how applicable the colour correction algorithm trained under one reflectance dataset is, to a scene consisting of reflectances from the other dataset? In other words, we look into the impact of reflectance dataset choices on the performance of colour correction algorithms.

For the list of the dataset we used, please refer to section 3.2. We also purposefully excluded the SFU combined reflectance dataset [97]. This is because the SFU reflectance dataset is a combination of multiple reflectance datasets, including the datasets listed above.

We study the relationship between reflectance datasets using different criteria, including: chromaticity distribution, colour solids formed by the reflectance dataset, the linear model basis for the reflectance dataset, and the performance of the colour correction matrices trained. Finally, we look into whether any of the metrics we explored has impacts on colour correction performances.

The conclusion we will take from this section is that the problem of finding a proper training dataset is far from being solved, and that none of the proposed metrics can really tell us about the ability of a dataset to be used as a training dataset. Also, we will see that there exists an exception: training datasets that present high regression residuals seem to be better for testing.

4.2 Chromaticity of the reflectance dataset under D65 illumination

We start our analysis on the reflectance datasets by qualitatively looking at the colours produced by them. More specifically, we achieve this by looking at the chromaticity distributions of the colours they produce.

We computed the chromaticities of the colours under D65 illumination in CIE 1976 Uniform Chromaticity Scale (CIE1976 UCS) [114, 115]. We chose the D65 illuminant, as it is one of the most commonly used illuminants in colour science studies. The CIE1976 UCS space was selected because colours are distributed more or less uniformly in this space. Specifically, the Euclidean distance between two colour points in this scale roughly corresponds to the perceptual difference between the two colours.

As stated in section section 3.3, each of the reflectance datasets is in the form of a $31 \times n$ matrix, where n is the number of reflectance spectra. Each reflectance spectrum in the dataset is sampled from 400 nm to 700 nm at 10 nm interval, with 31 samples in total.

We first compute the CIE XYZ tristimulus values for all reflectances in each of the reflectance datasets under D65 illumination, using the following equation:

$$\mathbf{Q} = \mathbf{X}^{\mathsf{T}}\mathbf{E}\mathbf{S}, \tag{4.1}$$

where $\mathbf{X}$ is a 31×3 matrix containing CIE XYZ colour matching functions, $\mathbf{E}$ is a 31×31 matrix containing the illuminant in the diagonal, and $\mathbf{S}$ is a $31 \times n$ matrix containing a set of n surface reflectance spectra.

The dimension of $\mathbf{Q}$ is therefore $3 \times n$. The rows of $\mathbf{Q}$ contain the X, Y and Z tristimulus values respectively. We then calculate the u', v' chromaticity values for each of tristimulus values. The u', v' values are obtained using the following formulae:

$$u' = \frac{4X}{X + 15Y + 3Z} \tag{4.2a}$$

$$v' = \frac{9Y}{X + 15Y + 3Z} \tag{4.2b}$$

We finally compute the mean u', v' chromaticities of the reflectance samples under D65 illumination for each reflectance dataset. These results are shown in Table 4.1. We can see in this Table that the mean u', v' values for each reflectance dataset are quite close to the D65 whitepoint. This means that these reflectance datasets conform to the *grey world* assumption [63]. For reflectance datasets that were created for covering a large gamut or for camera calibration (e.g. AGFA, Munsell, Macbeth and SG140), this is certainly a design choice. For reflectance datasets that sample real-world objects, this could be a confirmation of the grey world assumption, i.e. the average reflectance in the world is grey.

Table 4.1 The mean u', v' chromaticity values for various reflectance datasets

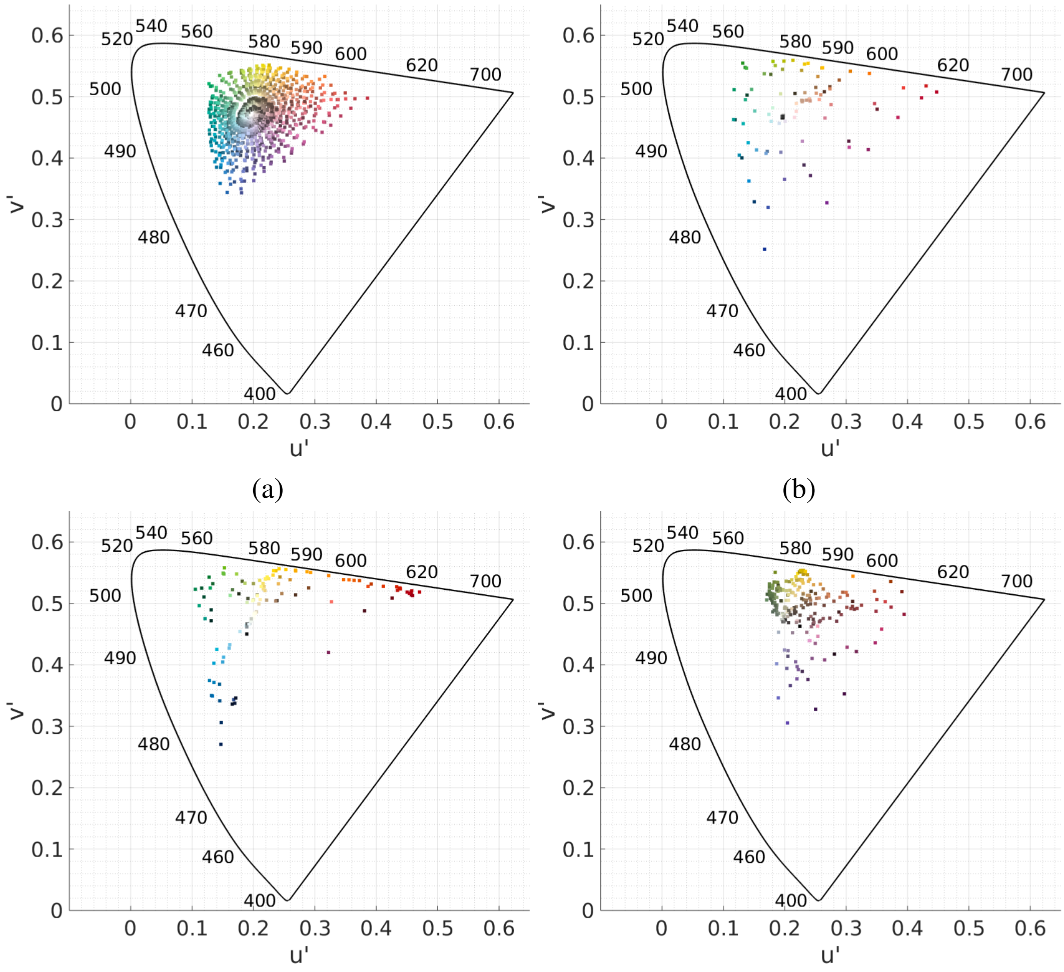
	u'	v'
AGFA	0.2050	0.4662
Natural	0.2287	0.4923
Munsell	0.2052	0.4712
SG140	0.2212	0.4767
Macbeth	0.2215	0.4648
Dupont	0.2338	0.4909
Object	0.2232	0.4935
D65 Whitepoint	0.1977	0.4685

Let us now dive deeper on the analysis of the u', v' chromaticities. In Figure 4.1 we plotted the u', v' chromaticity of each reflectance sample for four different datasets (Munsell, SG140, Dupont and Natural). In this figure, in addition to calculating the chromaticity of each reflectance sample, we also computed its sRGB value. We coloured each point in the plots using its associated sRGB value. It should be noted the colour for each point is for illustrative purpose only, and we cannot guarantee the accurate colour reproduction in both digital and printed format of this thesis.

We can see in this figure that the distribution of colours is different among the reflectance datasets. The chromaticity plot for the Munsell dataset shows that Munsell colour chips densely cover a relatively small colour space in a relatively uniform fashion. The chromaticity plot for SG140 shows that while this dataset covers a larger colour space, the sampling is more sparse and less uniform.

The u', v' chromaticity plot for Dupont dataset shows that this dataset is lacking in magenta coloured samples, while Natural dataset is lacking in samples that are deeply blue or deeply red. Natural dataset represents reflectances found in the nature, and for this reason it shows a dense sampling around the green/yellow and brown regions.

From these four examples, it is clear that there are important differences in the contents of the reflectance datasets, both in terms of the samples they contain, and in terms of their sampling density. In the next section, we will explore these differences further.



(a) (b)

(c) (d)

Fig. 4.1 UV chromaticity plot for (a) Munsell, (b) SG140, (c) Dupont, (d) Natural

4.3 Colour solid of the reflectance dataset under D65 illumination

In the previous section, we qualitatively looked at what reflectances are inside each dataset. In this section, we present a numeric metric for measuring the differences between reflectance datasets.

We start by treating the XYZ values for a reflectance dataset as points in 3D space, and we then calculate the convex hull for these points. We refer to the resulting construct as a *colour solid*. Formally, the convex hull $\mathrm{Conv}(\mathbf{X})$ for a set of points $\mathbf{X}$ is the smallest convex set that contains $\mathbf{X}$. In a convex set, any line segment $\overrightarrow{\mathbf{AB}}$ that joins any two arbitrary point $\mathbf{A},\mathbf{B}$ lines completely within the convex set itself. In 2D, a convex hull can be visualised as the shape enclosed by a rubber band around the point set **X**. An illustration of a convex hull in 2D is shown in Figure 4.2.

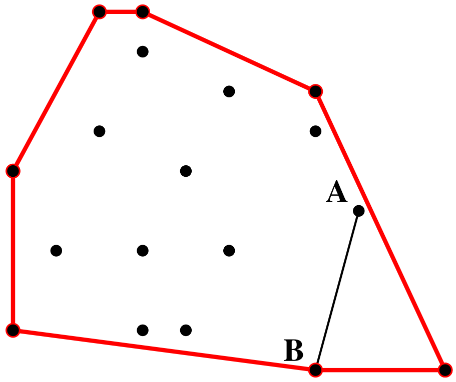


Fig. 4.2 A set of points in 2D, with their convex hull. The convex hull is the red line surrounding the points. The line $\overrightarrow{\mathbf{AB}}$ lies completely within the space enclosed by the red lines.

Let us note that by combining colours produced by the reflectance dataset, it is possible to generate colours that are within the convex hull which are not already produced by the reflectance dataset. This is because of the concept of additive colour mixing. This can be demonstrated by arranging tiles of colours in a mosaic pattern. When the observer is sufficiently far away from the mosaic pattern, due to the limited spatial resolution of the imaging sensor, it becomes impossible to distinguish individual tiles on the mosaic. This results in colour mixing. This is illustrated in Figure 4.3.

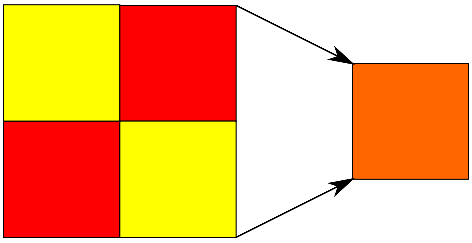


Fig. 4.3 When the observer is sufficiently away from a mosaic pattern, it becomes impossible to distinguish the individual tiles. This results in colour mixing.

For each reflectance dataset, we construct its *colour solid* (the convex hull). They are generated by applying the Quickhull algorithm on the XYZ values produced by each reflectance dataset [116]. As an example, Figure 4.4 shows the colour solid for Munsell dataset from 4 different perspectives. The surface of the colour solid is coloured based on the vertices of the convex hull. The individual dots within the colour solid represent individual reflectance spectrum. We will return to the idea of convex closure in the next chapter.

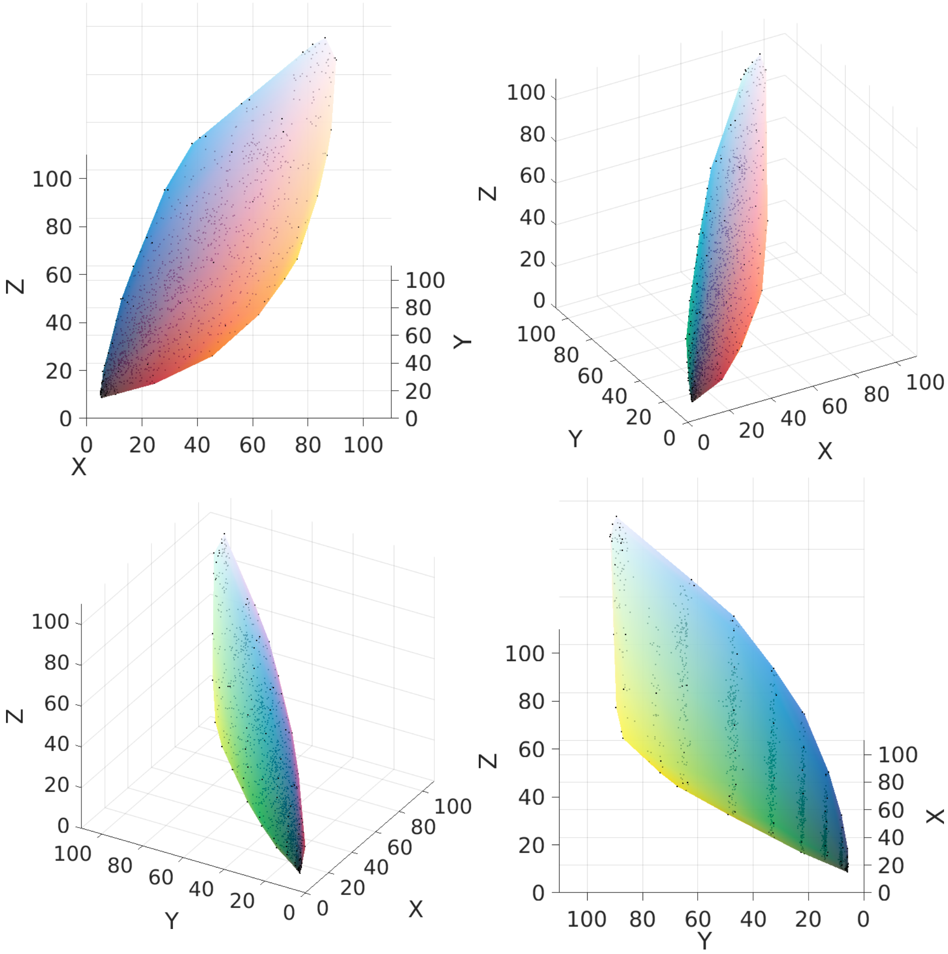


Fig. 4.4 The colour solid for Munsell reflectance dataset viewed from four different perspectives.

Once the *colour solids* are computed, we can perform pairwise comparisons to investigate how well one colour solid can be used to represent another colour solid. We devised a metric which we refer to as the *commonality ratio*.

4.3.1 Commonality ratio of the colour solid

Let us assume that we have two reflectance datasets – we train our colour correction algorithm on the training dataset, then apply the trained colour correction algorithm to the test dataset. How well will the trained colour correction algorithm perform in this scenario? In other words, how suitable is the training dataset for targetting this particular test dataset?

To answer these two questions, we need to revisit how colour correction works. In the simplest form of colour correction – linear least square colour correction, we train a 3×3 matrix by performing linear regression between device specific RGB values from the camera and their corresponding CIE XYZ tristimulus values. When the colour correction matrix is applied to new input RGBs, the residuals associated with the estimated XYZs are the smallest when the input RGBs are close to the RGBs within the training dataset.

In this subsection, we propose a quantitative metric which measures the suitability of a training reflectance set for a particular test reflectance dataset, by comparing their associated colour solids. We achieve this by calculating the ratio of the volume of intersection between the training and testing colour solids, and the volume testing colour solid. We refer to our metric as the *commonality ratio*.

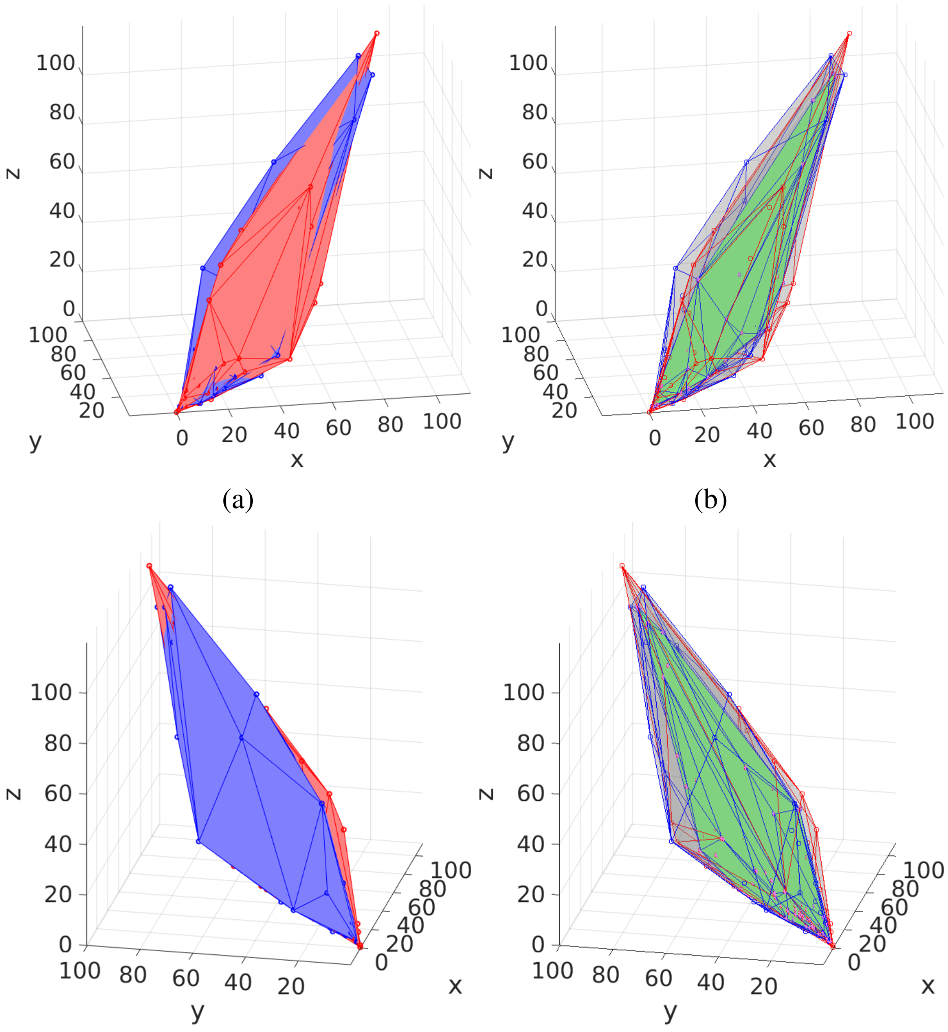
The commonality ratio, C can therefore be expressed using the following formula:

$$C(T,V) = \frac{T \cap V}{V}, \tag{4.3}$$

where T is the volume of the training colour solid, and V is the volume of the testing colour solid.

In short, our metric answers the question of how much the testing colour solid is within the training colour solid. By construction, if the training colour solid and the testing colour solids are identical, i.e. they occupy the same volume, the commonality ratio is 1. If there is no intersection between the two colour solid, then the ratio is 0. The commonality ratio is higher when a larger proportion testing colour solid resides within the training colour solid.

Figure 4.5 illustrates the ideas behind commonality ratio. It shows the colour solids of Natural and Dupont datasets drawn in the space, so they are intersecting, using two different perspectives. In Figure 4.5a and Figure 4.5c, the colour solid for Natural is coloured in red, while the colour solid for Dupont is coloured in blue. In Figure 4.5b and Figure 4.5d, we coloured the faces of both colour solids in semi-transparent light grey, leaving the wireframe of Natural as red, and the wireframe of Dupont as Blue. The intersecting volume between Natural and Dupont is coloured in green.



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

Fig. 4.5 The colour solids of Natural and Dupont drawn in the same space, so they are intersecting. (a) (b) and (c) (d) show the two colour solids in two different perspectives. In (a) and (c), Natural's colour solid is coloured in red, while Dupont's colour solid is coloured in blue. In (b) and (d), both Natural and Dupont's colour solids are coloured in transparent light grey, while their intersecting volume is coloured in green.

The commonality ratio is an asymmetric measure. The commonality ratio of Natural dataset to Dupont dataset is the volume of the green solid divided by the volume of the red solid, while for the Dupont dataset to Natural dataset, it is the volume of the green solid divided by the volume of the blue solid.

The commonality ratios between any two pairs of reflectance datasets were computed and they are presented in Table 4.2.

A case to remark is the SG140-Macbeth pair. The ratio of SG140 to Macbeth is 1.000, and for Macbeth to SG140 is 0.594. This is because Macbeth dataset is a strict subset of the SG140 dataset.

Munsell reflectance dataset achieves high average commonality ratio across all reflectance dataset. This is partially because Munsell dataset is the largest, and it covers the widest range. Conversely, when the Natural dataset is used as the training dataset, the commonality ratio tends to be lower. This may be potentially caused by the fact that the reflectances in the Natural dataset are restricted to samples collected from the nature, such as flower and leaves, while artificial paint and coloured chips are included in the other datasets.

Across the whole result table, the lowest commonality ratio (0.338) occurred when the training dataset is Natural, and when the testing dataset is Dupont. This is unsurprising, as the Dupont dataset consists of artificially coloured tiles, which do not exist in the Natural dataset. Furthermore, the Dupont dataset contain some pure red colours, which do not exist in the Natural dataset. This can be seen by looking at the UV chromaticity plots for Dupont and Natural in Figure 4.1c and Figure 4.1d respectively.

Other than the SG140-Macbeth pair, the highest commonality ratio (0.950) occurs between Munsell and Agfa. This is to be expected, as both datasets consists of colour resulted from artificial dyes, and they were both designed to cover a large range of colours.

Table 4.2 The commonality ratio between pairs of reflectance datasets

	Testing						
Training	AGFA	Natural	Munsell	SG140	Macbeth	Dupont	Object
AGFA	1.000	0.616	0.417	0.512	0.607	0.408	0.541
Natural	0.633	1.000	0.379	0.450	0.490	0.338	0.451
Munsell	0.950	0.839	1.000	0.830	0.846	0.794	0.861
SG140	0.793	0.677	0.564	1.000	1.000	0.622	0.725
Macbeth	0.558	0.439	0.342	0.594	1.000	0.419	0.527
Dupont	0.723	0.582	0.618	0.712	0.807	1.000	0.701
Object	0.703	0.570	0.492	0.608	0.744	0.514	1.000

4.4 Analysing the basis vectors for the reflectance dataset

In the previous section, we introduced the concept of colour solids. We measured the differences of the colour solids using commonality ratio, which measures the volumetric differences between them. In this section, we perform an analysis on the reflectance spectra themselves, by comparing the basis vectors associated with the reflectance spectra of each dataset.

This section is organised as follows: we start by discussing how the basis vectors and the coefficient matrix are obtained. We then adapt the concept of the Vora values [117] to use it for comparing reflectance datasets, and we finally discuss the variance of the reflectance dataset that can be captured by the first few basis vectors.

Let us first revisit how reflectance spectra and reflectance datasets are represented. A single reflectance spectrum can be represented by a 31×1 column vector $\mathbf{s}$. A reflectance dataset is constructed by concatenating n reflectance spectrum vectors horizontally, resulting a single $31 \times n$ matrix, that we denote as $\mathbf{S}$.

We can represent $\mathbf{S}$ as a linear combination of bases and coefficients:

$$\mathbf{S} = \mathbf{UK}, \tag{4.4}$$

where:

- $\mathbf{U}$ is a 31×31 orthonormal matrix containing the basis vectors, where each column of the matrix is a basis vector;
- $\mathbf{K}$ is a $31 \times n$ matrix containing the coefficients. Each column contains the coefficients associated with a single reflectance sample.

4.4.1 Obtaining the basis for the linear combination

To obtain the basis for the linear combination, we start by decomposing $\mathbf{S}$ using singular value decomposition:

$$\mathbf{S} = \mathbf{UDV}^{\mathsf{T}}, \tag{4.5}$$

where:

- $\mathbf{U}$ is a 31×31 orthonormal matrix containing the basis for the column space;
- $\mathbf{D}$ is a $31 \times n$ matrix with non-negative real numbers on the diagonal, these numbers are known as the singular values;
- $\mathbf{V}$ is a $n \times n$ orthonormal matrix containing the basis for the row space.

Therefore, $\mathbf{U}$ contains the basis vectors associated with the reflectance dataset $\mathbf{S}$. Each column of $\mathbf{U}$ is a single orthonormal basis vector. They are also known as the principal components. Each basis vector is associated with a singular value, which is a single element in the main diagonal of **D**. The singular value indicates the amount of variance of the dataset that is accounted for by the basis vector. Typically, the singular values in **S**, and therefore the basis vectors in **U** are sorted in descending order, although this depends on the exact algorithm or implementation used for decomposition.

We can reconstruct **S** with a high degree of accuracy without using all of the available bases. It is sufficient to use bases that are associated with large singular values, as they capture large amounts of variance. Typically, only 3 to 6 basis vectors are sufficient for reconstructing **S** to an acceptable degree of accuracy, as they are able to capture adequate amount of variance from **S** [118]. However, to achieve perfect reconstruction, every basis vector in **U** is required.

Typically the basis vectors that account for the most variance are chosen. Representing a dataset without using all of the available basis vectors is a form of dimensionality reduction [119]. Assuming that the basis vectors in **U** are sorted in a descending order based on the amount of variance they capture, typically only the first few vectors in **U** are chosen to form the basis matrix, which we denote using $\widetilde{\mathbf{U}}$.

4.4.2 The Reflectance Vora Values between datasets

In this subsection, we measure the difference between the bases of the reflectance datasets using the Vora value [117].

The Vora value is a metric originally designed to measure how well a camera sensor set measures colour and is related to the method of principal angles. It measures the principal angles between the two sub-spaces spanned by the camera spectral sensitivity functions and the colour matching functions. For a conventional set of camera spectral sensitivities comprising 3 filters, the sub-space spanned by these three spectra is three dimensional. The same can be said about the sub-space spanned by the colour matching functions. Hence, we have three, in general non-zero, angles (called principal or canonical angles) that minimise the ‘gap’ between the two sub-spaces. Vora value is the sum of squared cosines of the three principal angles normalised by the number of the principal angles (here 3). The value tends to 1 for colorimetric sets of sensors; and conversely, tends to 0 for sets of sensors which are poor from the point of view of colour measurement. An imaging sensor set with a Vora value of 1 is a perfect colorimeter i.e. it measures colour equally well as the colour matching functions do. Practically, the colour reproduction errors arising from an imaging sensor with a Vora value of 0.9 are perceptually tolerable [120].

The formula for Vora value $V(\mathbf{X},\mathbf{R})$ is defined as:

$$V(\mathbf{X},\mathbf{R}) = \frac{\mathrm{Tr}(\mathbf{X}\mathbf{X}^{+}\mathbf{R}\mathbf{R}^{+})}{\alpha}, \tag{4.6}$$

where $\mathbf{X}$ is the matrix containing the colour matching functions, and $\mathbf{R}$ is the matrix containing the camera spectral sensitivity functions, and α is the rank of $\mathbf{X}$ [121]. $\mathrm{Tr}(\bullet)$ represents the trace of a matrix, which is a sum of the elements on the main diagonal of a matrix.

Equation 4.6 can be rewritten as:

$$V(\mathbf{N},\mathbf{O}) = \frac{\mathrm{Tr}(\mathbf{N}\mathbf{N}^{\mathbf{T}}\mathbf{O}\mathbf{O}^{\mathbf{T}})}{\alpha}, \tag{4.7}$$

where $\mathbf{N}$ and $\mathbf{O}$ are the matrices containing the orthonormal basis of the colour matching functions and the camera spectral sensitivities respectively (the pseudo inverse of an orthonormal matrix is the matrix transpose).

Here, we adapted the Vora value formula for measuring the difference between the bases of reflectance datasets. We refer to our metric as the Reflectance Vora Value. We substitute $\mathbf{N}$ and $\mathbf{O}$ for the orthonormal bases of the two reflectance datasets. We denote the basis of one of the reflectance datasets as $\widetilde{\mathbf{U}}_1$, and the basis of the other reflectance dataset as $\widetilde{\mathbf{U}}_2$. The formula for reflectance Vora value between two reflectance datasets, $V_{\mathrm{ref}}(\widetilde{\mathbf{U}}_1,\widetilde{\mathbf{U}}_2)$ is defined as:

$$V_{\mathrm{ref}}(\widetilde{\mathbf{U}}_{\mathbf{1}},\widetilde{\mathbf{U}}_{\mathbf{2}}) = \frac{\mathrm{Tr}(\widetilde{\mathbf{U}}_{\mathbf{1}}\widetilde{\mathbf{U}}_{\mathbf{1}}^{\mathbf{T}}\widetilde{\mathbf{U}}_{\mathbf{2}}\widetilde{\mathbf{U}}_{\mathbf{2}}^{\mathbf{T}})}{\beta}, \tag{4.8}$$

where β is the rank of $\widetilde{\mathbf{U}}_1$ and $\widetilde{\mathbf{U}}_2$. We are assuming that $\widetilde{\mathbf{U}}_1$ and $\widetilde{\mathbf{U}}_2$ have the same rank. The rank of $\widetilde{\mathbf{U}}_1$ or $\widetilde{\mathbf{U}}_2$ is also known as the dimension of the basis.

We also expect $V(\widetilde{\mathbf{U}}_{\mathbf{1}},\widetilde{\mathbf{U}}_{\mathbf{2}}) = V(\widetilde{\mathbf{U}}_{\mathbf{2}},\widetilde{\mathbf{U}}_{\mathbf{1}})$, due to the mathematical property of $\mathrm{Tr}(\mathbf{AB}) = \mathrm{Tr}(\mathbf{BA})$. This means given that $\widetilde{\mathbf{U}}_1$ is generated from reflectance dataset $\mathbf{S}_1$ and $\widetilde{\mathbf{U}}_2$ is generated from reflectance dataset $\mathbf{S}_2$, based on our metric, reconstructing $\mathbf{S}_1$ using $\widetilde{\mathbf{U}}_2$ has the same accuracy as reconstructing $\mathbf{S}_2$ using $\widetilde{\mathbf{U}}_1$.

For each pair of the reflectance datasets, we computed its Reflectance Vora value for the bases of the dimensionality ranging from 3 to 31. We computed the mean Vora values of all pairs, and plotted them in Figure 4.6. The reason we chose to start from 3 dimensions is because previous research had suggested that the minimum number of dimensions required to reproduce a reflectance dataset to an acceptable degree of accuracy is 3 [122, 118, 123, 124, 96].

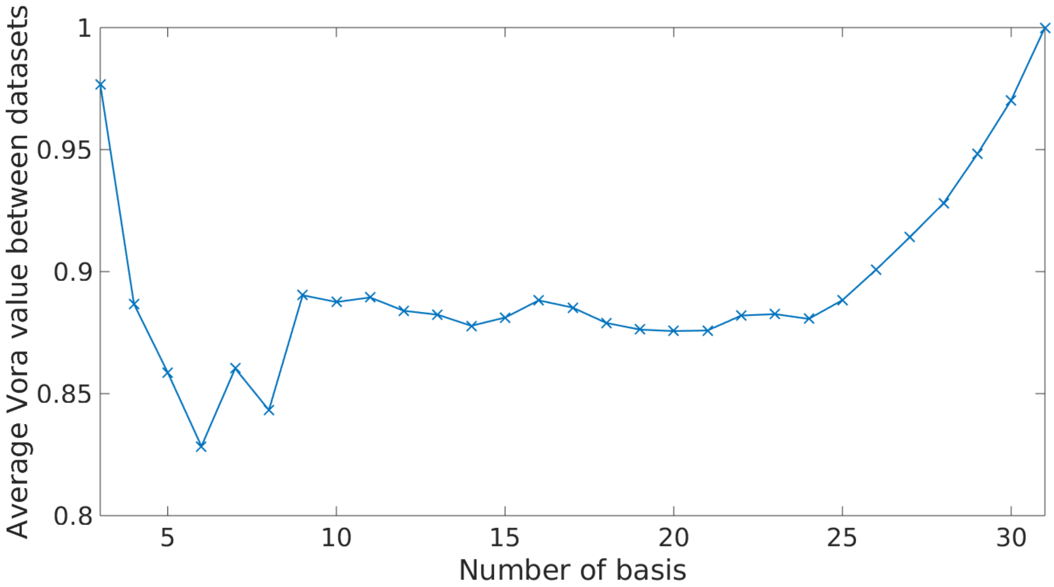


Fig. 4.6 A line plot showing the mean reflectance Vora value across all datasets against the dimension of the basis, for dimensions 3 to 31.

Figure 4.6 shows that the mean Vora value between reflectance dataset is above 0.95 for three dimensional basis. It rapidly drops to below 0.85, for six dimensional basis. This happens due to "underfitting" when the number of dimensions is below 6. The reconstructed reflectances lack details, so they look very similar across different datasets. The mean Vora value gradually climbs up to just below 0.90 when the number of dimensions reaches 9, and it plateaus until the number of dimensions is 24. This suggests that the Reflectance Vora values calculated for 9-24 dimensions are a fair representation of the similarity between reflectance datasets. From 24 to 31 dimensional basis, the mean Vora value gradually increases to 1. This happens due to "overfitting". With the addition of basis of low importance, it becomes easier to reconstruct one reflectance dataset using the basis from another reflectance dataset.

In, Table 4.3, we show the reflectance Vora values for each pairs of datasets represented with 9-dimensional basis. We have chosen 9-dimensional basis, because previous research suggested that 9-dimensional basis are sufficient for reconstructing reflectances to a high degree of accuracy. Additionally, Figure 4.6 shows that reflectance Vora values do not change significantly for 9-24 dimensional basis.

Table 4.3 The Reflectance Vora Values for each pair of the reflectance datasets with 9-dimensional basis

	AGFA	Natural	Munsell	SG140	Macbeth	Dupont	Object
AGFA	1.000	0.859	0.913	0.919	0.939	0.902	0.897
Natural	0.859	1.000	0.808	0.809	0.812	0.786	0.875
Munsell	0.913	0.808	1.000	0.961	0.931	0.955	0.854
SG140	0.919	0.809	0.961	1.000	0.957	0.944	0.899
Macbeth	0.939	0.812	0.931	0.957	1.000	0.924	0.897
Dupont	0.902	0.786	0.955	0.944	0.924	1.000	0.857
Object	0.897	0.875	0.854	0.899	0.897	0.857	1.000

Table 4.3 shows symmetry along the diagonal – the lower triangular of the table is the mirror of the upper triangular. This is to be expected, due to the construction of Equation 4.8. The main diagonal of the table has values of 1 as comparing the identical bases must give a reflectance Vora value of 1.

Table 4.3 shows a similar pattern to Table 4.2. The lowest reflectance Vora value happens between Dupont dataset and Natural dataset. This is unsurprising, as Dupont dataset consists of reflectance spectra from artificially coloured tiles, while the Natural dataset consists of reflectance spectra from flowers, leaves and other colourful plants. The reflectance spectra in these two datasets are very different which means that their bases should be also very different.

The highest reflectance Vora value occurs between Munsell dataset and SG140 dataset. This is probably because both datasets comprise tiles coloured by artificial dye, and SG140 dataset was designed to cover a wide range of colours. Additionally, reflectance dataset pairs involving Munsell dataset have high reflectance Vora value in general. This is probably due to the large size of the Munsell dataset – it has the widest range of reflectance spectrum samples.

4.4.3 Variance of the dataset captured by the basis vectors

Individual reflectances from a dataset are reconstructed by linearly combining the basis vectors. The accuracy of the reconstruction can be improved by using more basis vectors. The limit to the accuracy of the reconstruction depends on the amount of variance captured by the chosen basis vectors.

As stated above, the singular value associated with a basis vector indicates the amount of variance that the basis vector accounts for. The percentage of variance captured by the

i^{th} basis vector, p_i can be calculated using the singular value matrix $\mathbf{D}$, using the following formula:

$$p_i = 100 \times \frac{D_{ii}^2}{\sum_{j=1}^{n} D_{jj}^2} \,. \tag{4.9}$$

The cumulative percentage of variance captured by the first i^{th} basis vectors, c_i can be calculated using summing up the first p_i terms:

$$c_i = 100 \times \sum_{j=1}^{i} p_i = \frac{\sum_{k=1}^{i} D_{kk}^2}{\sum_{j=1}^{n} D_{jj}^2} \,. \tag{4.10}$$

Our analysis focus on the first six basis vectors, as they are sufficient to reconstruct reflectances within a dataset to a high degree of accuracy.

The result of our analysis is shown in Table 4.4. In this table, in the first 6 columns, we show the percentage of variance captured by basis 1 to basis 6 individually. In the last two columns, we show the percentage of variance captured by the first three basis and the first six basis. The results show that for each reflectance dataset, more than 98.1% of the total variance can be captured by the first three basis vectors, and more than 99.6% can be captured by the first six basis vectors. This suggests that 3-6 basis vectors are sufficient for reconstructing the reflectances to a high degree of accuracy. This is consistent with the findings of [122, 118, 123, 124, 96].

Table 4.4 The percentage of variance captured by the basis vectors for each reflectance dataset

	Basis							
Dataset	1	2	3	4	5	6	1-3	1-6
AGFA	93.156	5.130	1.422	0.114	0.096	0.045	99.708	99.964
Natural	90.763	4.157	3.277	0.862	0.439	0.181	98.197	99.680
Munsell	92.161	5.328	2.011	0.258	0.124	0.041	99.500	99.922
SG140	86.237	9.789	3.094	0.455	0.257	0.059	99.119	99.889
Macbeth	84.372	11.075	3.657	0.491	0.209	0.073	99.105	99.877
Dupont	87.782	9.543	2.156	0.324	0.099	0.043	99.482	99.948
Object	92.857	4.643	1.638	0.351	0.227	0.129	99.137	99.844

4.5 Colour correction experiments

So far we investigated the differences between reflectance datasets by exploring their chromaticity distributions, the intersection of their colour solids, and the reflectance Vora value for the basis vectors. However, it is not clear if the differences between reflectance datasets have any practical consequences. In this section, we answer the question if the choice of the training reflectance dataset has impact on the performance of colour correction.

To measure the impact of the reflectance dataset choice on colour correction, we perform a colour correction experiment using the linear least square (LSS) algorithm under D65 illumination, considering the 28 camera dataset [88] . We selected the LSS method as it is the simplest and most popular method for colour correction.

Let us recall here the formulation of LLS. In LLS colour correction, we look for the 3×3 matrix M that minimises:

$$\min_{\mathbf{M}}\{||\mathbf{MP}-\mathbf{Q}||_F\} \ , \tag{4.11}$$

where **P** and **Q** are respectively the colour values for the camera sensors and the XYZ tristimulus values, respectively. The $||\cdot||_F$ above denotes Frobenius norm.

The most common solution for this problem – the ordinary least-squares 3×3 regression matrix **M** can be solved by Moore-Penrose pseudoinverse:

$$\mathbf{M} = \mathbf{QP}^{\mathsf{T}}(\mathbf{PP}^{\mathsf{T}})^{-1}. \tag{4.12}$$

Therefore, the colour correction matrix **M** can be understood as a matrix that best maps the the device specific RGBs captured by the camera to their corresponding CIE XYZ tristimulues values in the ordinary least-square sense, for a particular dataset.

To perform our colour correction experiment, we repeated the following procedure for each of the cameras considered. For each reflectance dataset we have, we trained a colour correction matrix. We then applied the colour correction matrix to the device specific RGBs generated from the other reflectance datasets, in order to estimate the corresponding CIE XYZs. We then measured the colour differences between the ground truth XYZs and the estimated XYZs using the CIE 1976 Colour Difference Formula [125] (CIELAB ΔE^*_{ab}). Finally, we calculated the mean across all sample spectra and all the different cameras.

In other words, we validate the colour correction matrices trained on one dataset by applying them to device specific RGBs generated from the other datasets. Let us give an example of the previous explanation. When we trained the colour correction matrices under AGFA dataset, we tested them on Natural, Munsell, SG140, etc. Therefore, in this scenario we investigate how applicable is a colour correction matrix learnt for a specific dataset to other reflectance datasets.

The mean CIELAB ΔE^*_{ab} for this experiment are shown in Table 4.5. The datasets which produce the highest and lowest mean CIELAB ΔE^*_{ab} are highlighted in the table.

Table 4.5 The mean CIELAB ΔE for applying colour correction matrices to RGBs generated from different reflectance datasets

		Testing							
		AGFA	Natural	Munsell	SG140	Macbeth	Dupont	Object	Mean
Training	AGFA	-	2.48	1.95	3.76	2.93	4.83	2.17	**3.02**
	Natural	1.48	-	1.72	3.01	2.65	3.84	1.81	2.42
	Munsell	1.37	2.49	-	3.54	2.96	4.62	2.20	2.87
	SG140	1.40	1.91	1.71	-	2.80	3.48	1.64	2.16
	Macbeth	1.38	2.14	1.79	3.23	-	4.15	1.83	2.42
	Dupont	1.41	1.84	1.79	2.74	2.73	-	1.55	**2.01**
	Object	1.36	2.17	1.94	3.18	2.75	3.87	-	2.55
	Mean	**1.40**	2.17	1.82	3.24	2.80	**4.13**	1.87	-

Table 4.5 shows that the choice of the training dataset has an impact on the performance of the resulting colour correction matrices.When colour correction matrices are trained using a single reflectance dataset and then tested in all the other datasets, the reflectance dataset that produces the highest mean CIELAB ΔE^*_{ab} is AGFA (3.02), while Dupont produced the lowest mean (2.01). This suggests that Dupont is the best dataset for training colour correction matrices, while AGFA is the worst.

Table 4.5 also shows that the testing dataset choice can also affect the process of evaluating colour correction algorithms. When a single reflectance dataset is used to test colour correction matrices trained by all other datasets, Dupont dataset produced the highest mean CIELAB ΔE^*_{ab} (4.13), while AGFA produced the lowest CIELAB ΔE^*_{ab} (1.40). This suggests that Dupont dataset is the most challenging dataset for testing colour correction matrices, while AGFA is the least challenging.

From the previous results is interesting to note that the most challenging testing dataset appears to be the best training dataset. This suggests that it is beneficial to incorporate challenging reflectance samples in the process of training colour correction algorithm.

The previous conclusion made also look at a second scenario, in which we apply the colour correction matrices to the RGBs generated from the training dataset itself. In this scenario, we effectively measure the regression residuals of training the colour correction matrix. The regression residuals of a variable are the differences between the estimated values and the actual observed values. The mean regression residual of a variable is a good indication of how varied the observations of that variable are. In this case we also use mean CIELAB ΔE^*_{ab} as an indicator for how varied the reflectance spectra are within a dataset.

Table 4.6 present the results for this experiment. This table shows that the reflectance dataset that gives the highest regression residual is the Dupont, while AGFA dataset gives the lowest regression residual.

Table 4.6 The mean CIELAB ΔE for applying colour correction matrices to device specific RGBs generated from the training dataset

AGFA	Natural	Munsell	SG140	Macbeth	Dupont	Object
1.09	1.70	1.45	2.86	2.57	**3.13**	1.74

Interestingly, comparing Tables 4.5 and 4.6 it seems that a reflectance dataset with a high regression residual appears to perform better as a training dataset. We look further at this relation in Figure 4.7. The y-axis of the figure has the mean CIELAB ΔE^*_{ab} values when using the colour correction matrices that were trained with other reflectance datasets. The x-axis presents the mean CIELAB ΔE^*_{ab} value when using the colour correction matrices that were trained with same dataset (i.e. the regression residuals). The Pearson's r score between the the variables at y-axis and x-axis is -0.92, which (as we were hypothesizing) indicates a strong negative correlation between the regression residuals of a reflectance dataset, and how well this reflectance dataset performs as a training dataset.

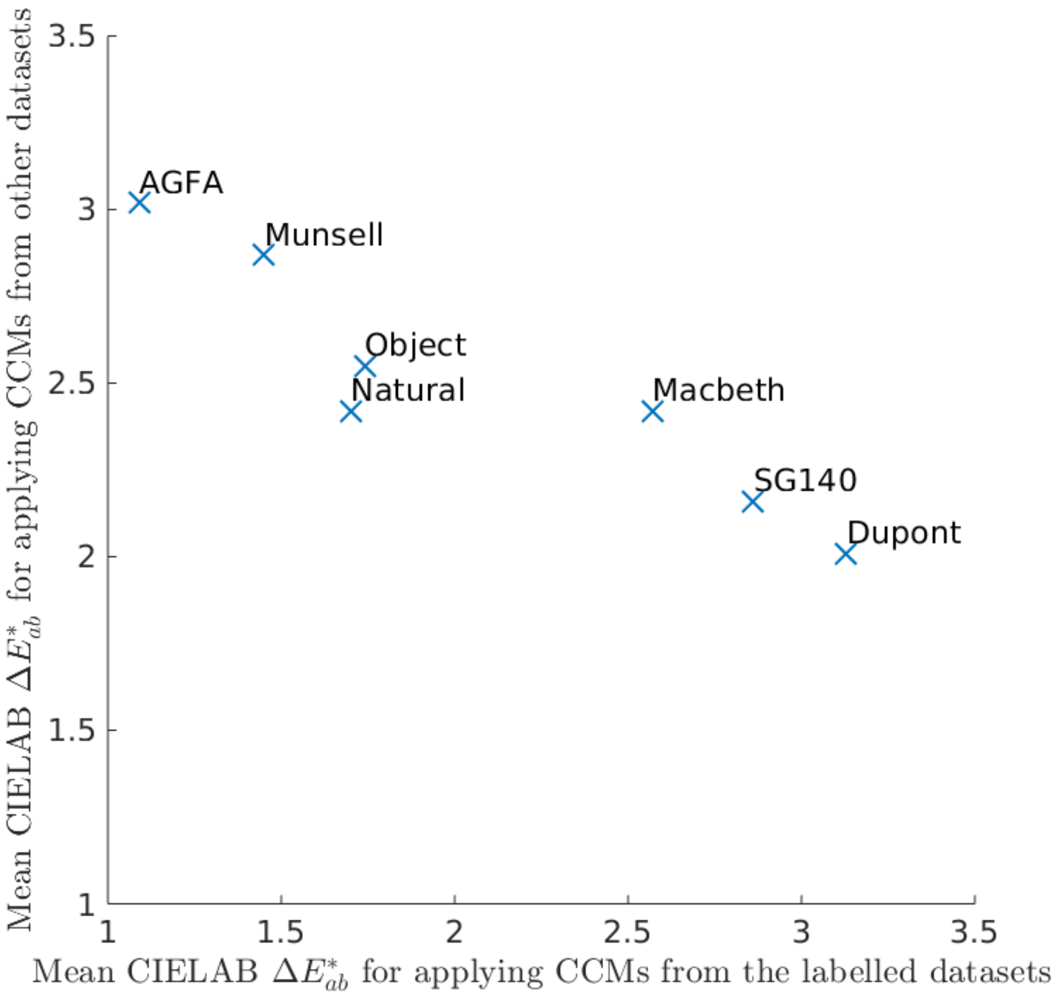


Fig. 4.7 Scatter plot showing the mean CIELAB ΔE^*_{ab} of applying colour correction matrices (CCMs) on RGBs generated from different reflectance datasets and the training dataset (r = -0.92)

4.6 The relationship between colour correction experiments and other metrics

In this chapter, we have performed pairwise comparisons of different reflectance datasets using various metrics. Now we look at whether there is a link between the metrics we investigated, and the performance of colour correction experiments.

In particular, we want to investigate whether the various metrics introduced have any correlation with the CIELAB ΔE^*_{ab} values obtained in the last section. To this end, we computed the Pearson correlation coefficient (Pearson's r) between the chosen metric and the CIELAB ΔE^*_{ab} . We then visualise the relationship between using scatter plots.

In Figure 4.8, we plotted the mean CIELAB ΔE^*_{ab} against the Vora value between the training and testing reflectance dataset. We showed that the they have no correlation, as the Pearson's r of 0.01. Figure 4.9 shows that the ratio of the intersecting convex hull volume has no impact on CIELAB ΔE^*_{ab} , with a Pearson's r of -0.10.

The results in this section show that our proposed metrics are not useful in order to predict the colour correction performance of a training reflectance dataset.

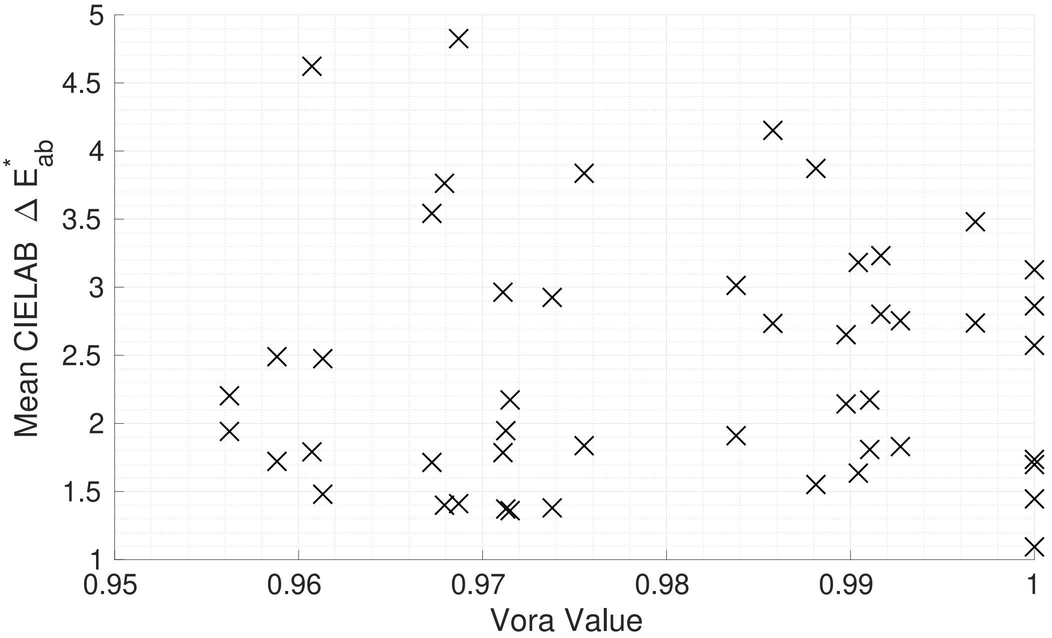


Fig. 4.8 A scatter plot showing mean CIELAB ΔE^*_{ab} against Vora values between training and testing reflectance datasets ($r = 0.01$)

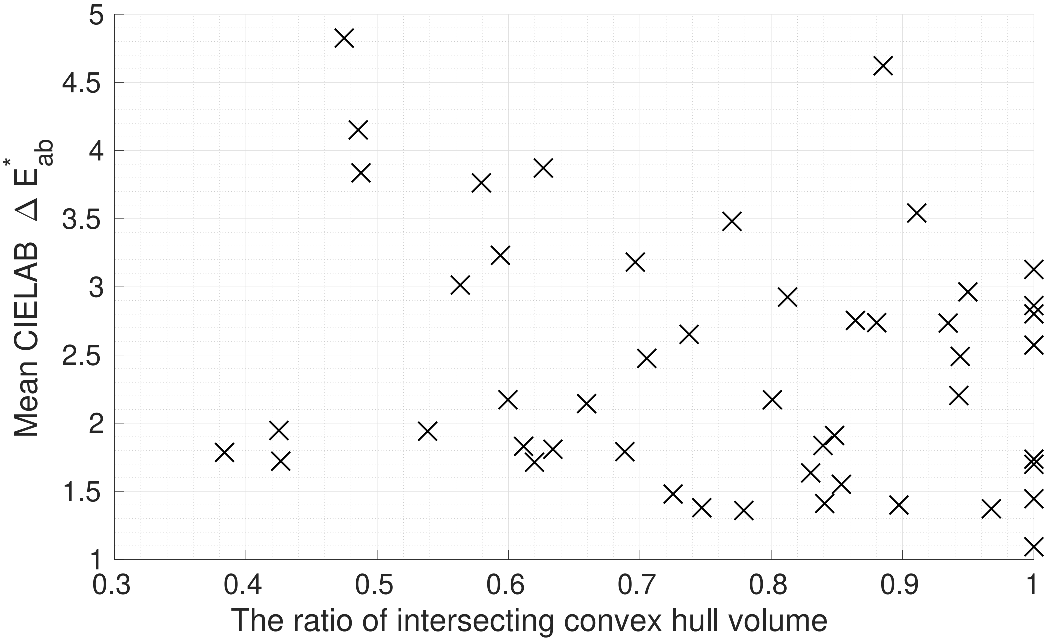


Fig. 4.9 A scatter plot showing mean CIELAB ΔE^*_{ab} against against the ratio of convex hull intersection between the training and testing reflectance datasets ($r = -0.10$)

4.7 Conclusion

In this chapter, we started by computing chromaticities in the CIE 1976 uniform chromaticity scale diagram for various reflectance datasets under D65 illumination. This allowed us to perform a qualitative visual comparison of the colours in each dataset. These plots show that the reflectance datasets are quite different from one another. We then generated colour solids for the reflectance datasets under D65 illumination and proposed and computed the contribution ratio for the colour solids. We then looked into the basis functions for the reflectance datasets. From them, we proposed a modification of the Vora Value measure, in order to deal with the comparison of reflectance datasets.

We then looked at how the Linear Least-Squares Colour Correction Method is affected by the training dataset, and show that it exists a negative correlation between the regression residuals of a reflectance dataset, and how well this reflectance dataset performs as a training dataset.

As a common thread through all our results is that the choice of dataset *matters*. The reflectance sets are different from one another. And, adopting any as a training set impacts on the colour correction performance of another set (used for testing). I doubt any researchers would have predicted that the Dupont reflectance set – which itself is the most difficult to colour correct – would be the one that is best for training colour correction in the sense that the trained correction matrices generalise to other datasets.

Chapter 5

Integrating the space of reflectance spectra

5.1 Introduction

The content of this chapter is based on a published conference paper [126].

Unsurprisingly, many colour algorithms ranging from colour correction to colour constancy to spectral estimation are developed from and tuned using measured reflectance data. Because the spectral sensitivities of the camera are different (not a linear transform) from the human vision system sensitivities there is no way to exactly solve the colour correction problem. *Metamerism* occurs as a natural consequence of the difference in spectral sensitivities leads. Metamerism [127–129] is the circumstance where two spectra look different, integrate to different RGBs for the camera, but are identical to the human observer and vice versa. Clearly, we cannot map the same camera RGB to different sRGB colour coordinates, so error free colour correction is impossible. Colour correction algorithms dually seek both to fit training data and to generalise in the sense that they colour correct the RGBs for unseen reflectances with low error.

The role of the reflectance 'training data' is very important. Any regression-based algorithm (e.g. simple least-squares) is guaranteed to be optimal for the training dataset but may be sub-optimal for other sets of reflectances, i.e. for the other colours that *did not* appear in the training dataset. The performance of colour correction algorithm depends on the training reflectance dataset. Consequently, when comparing colour algorithms, the choice of reflectance datasets affects the resulting rankings.

In this chapter, we are going to consider the *reflectance dependency* problem in detail. We will begin by making the observation that reflectances are high dimensional data points and if we wish to sample the world and arrive at a representative sample-set then we must address the issue of high dimensionality.

There are a couple of important questions to consider here. First, to what extent is it possible to decide *a priori* the distributions of reflectances in the world. Clearly, if we are interested in colour with respect to a restricted environment – where not all colours naturally appear – then it makes sense to use only a subset of all plausible colours. We will not consider this scenario here. Indeed, we will always assume that all typical photographic scenes might be encountered. Second, once the environment – or range of colours – is defined, the pertinent question to ask is how many reflectances do we need to sample to represent the space of all reflectances?

Let us consider this second question. We are going to carry out an abstract experiment which, at least initially, will seem unrelated to the problem at hand. We ask the question: "how many points do we need to generate – uniformly and randomly in a unit square – to 'cover' the square?" We then extend this question to hypercubes of arbitrary dimension. In providing an answer, we will come to understand a central problem of sampling: the need for many points. Usefully, it will become apparent that the idea of a hypercube – and how it is sampled – is key to to the arguments we will develop in this chapter.

In Figure 5.1, we plot 100 uniformly and randomly generated points in the unit cube (blue crosses). The unit square looks like it is well covered, the points seem to fill the square. To more formally measure the point coverage, we calculate the convex hull of the points, bounded by the red line in Figure 5.2. We say that the convex hull bounds the convex closure of the point set. The area of the convex hull is just 0.83. For this experiment, the 100 random points results covers 83% of the area of the unit square.

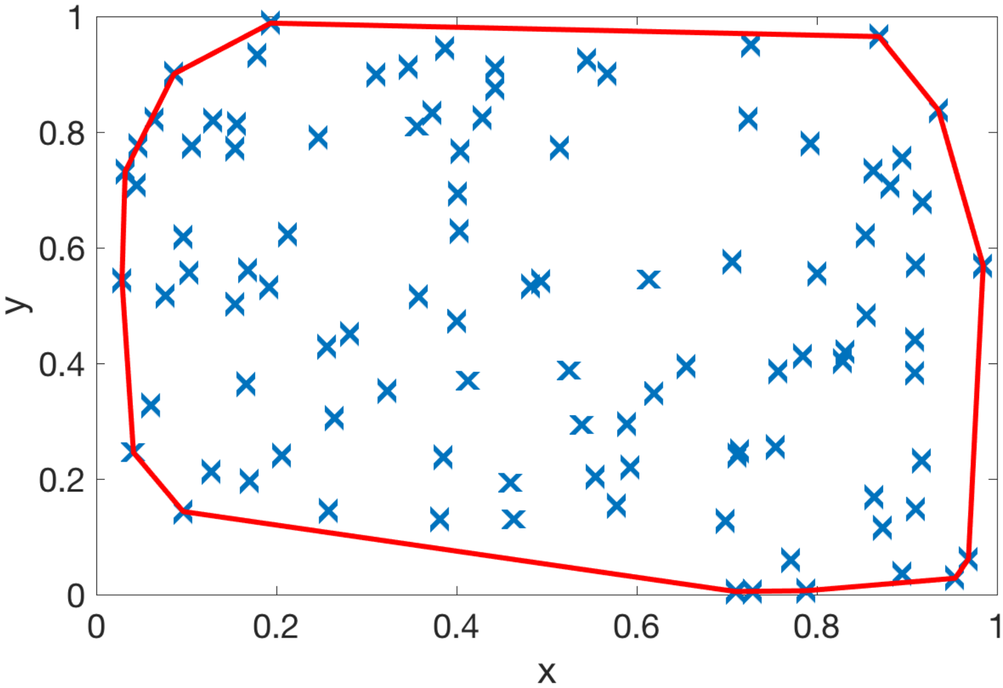


Fig. 5.1 100 points are uniformly and randomly selected and plotted (blue crosses). The convex hull of these points is shown in red.

Now, let us move from a 2-dimensional unit square to a d-dimensional unit cube. We know that square, cube or d-dimensional hypercube have an area, volume or hyper-volume of 1 by definition. As before, for a given d-dimensional space, we randomly sample the unit hyper-cube, calculate the convex hull of the points and then we compute the corresponding volume. We ask: "how many points do we need to randomly sample in order to approximate the volume?" For the purposes is deemed to be 'good enough'.

In Figure 5.2, we plot log_{10}(number of points) on the x axis. For a given hypercube we randomly select 100, 1000, 10000, 100,000 and 1 million points) and then calculate the convex hull of the points and their volumes. We plot log_{10}(number of points) against volume estimated for 3, 4, 5 and 6 dimensional data.

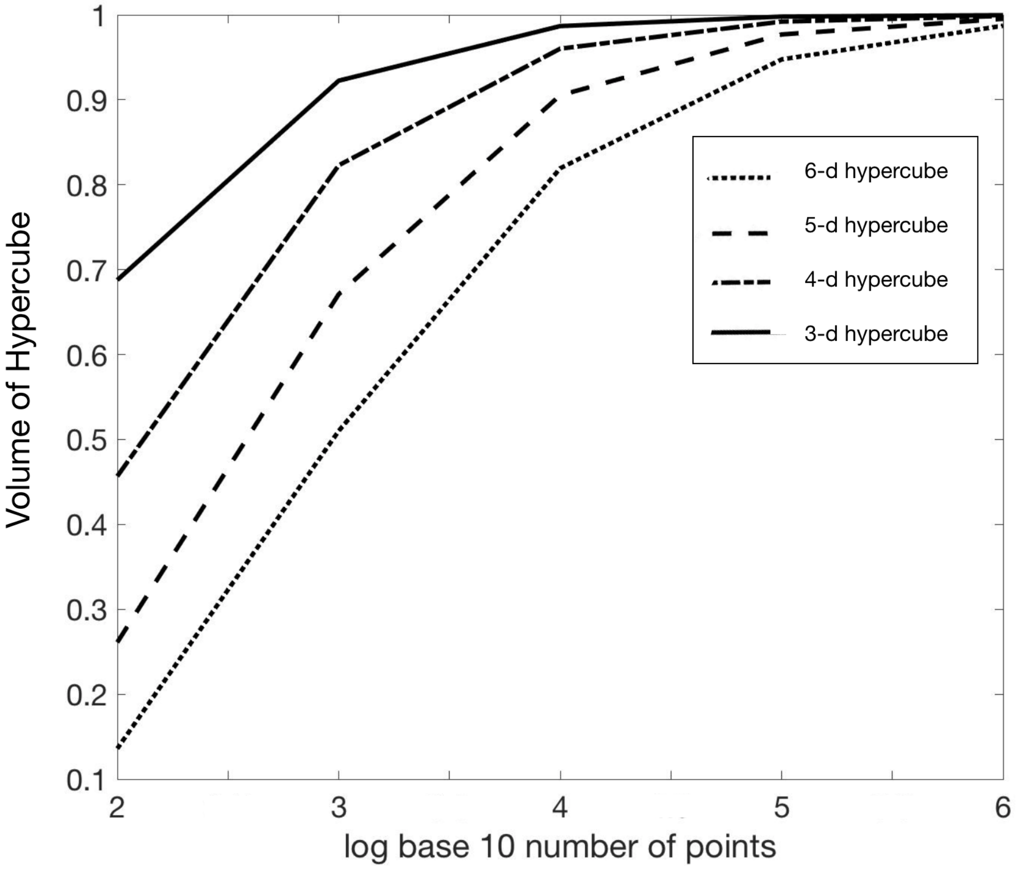


Fig. 5.2 Volume of the convex hull of uniformly and randomly selected points in dimensions 3, 4, 5 and 6. To cover 99% of the hypercube we need $O(10^d)$ points.

Very roughly, our experiment shows that to cover 99% of the unit volume (independent of the hypercube dimension), we need $10^{\mathrm{d}+1}$ points, where d is the dimension of the hypercube. This shows that the number of sample points needed to cover a hypercube grows exponentially with respect to the dimension of the hypercube. This simple empirical observation turns out to be true in the limit (where the dimension becomes large) [130].

Let us return to the problem at hand: representing reflectance data. Spectra are often represented by 31 measurements: every 10 nanometres across the visible spectrum (400 to 700 nanometres). Spectral datasets range from a few dozens of reflectance spectra measurements to a few thousands, e.g. the Macbeth ColorChecker (24 reflectances) [95], the Dupont reflectance set (120 reflectances), the object dataset (170 reflectances) [96] and the SFU composite reflectance set [97] (2000 reflectances including the Macbeth and Object dataset). All these datasets have been widely deployed in the development of colour imaging algorithms. Given these data sets comprise points in 31-dimensional space, they are modest in size. Can even 2000 reflectances adequately represent all reflectances we might encounter in the natural world?.

In this chapter, we propose that the problem lies not in the dataset – and the number of sample points – *per se* but rather than how those samples are used. We begin by drawing a simple lesson from the apparently abstract example of Figure 5.1 and Figure 5.2. There, we sought to make enough samples so that the volume of the enclosing convex hull of the point set approximated the volume of the unit-cube. In terms of reflectances, this convex hull idea, where all reflectances therein are equally likely, is not only useful for our thought experiment but actually reflects the physics of image formation.

Suppose we have a surface that is a patchwork of materials (e.g. a simple coloured texture) and we view this texture from far enough away. The effective reflectance we see is a combination of the underlying reflectances. In other words, mathematically, the combined reflectances is a convex combination of the individual reflectances (in proportion to the area that each reflectance covers). By making mosaics of colours by taking the reflectances in a given sampled data set we can – when viewing the mosaic from far enough away – make every reflectance that lies in the convex closure of the reflectance dataset.

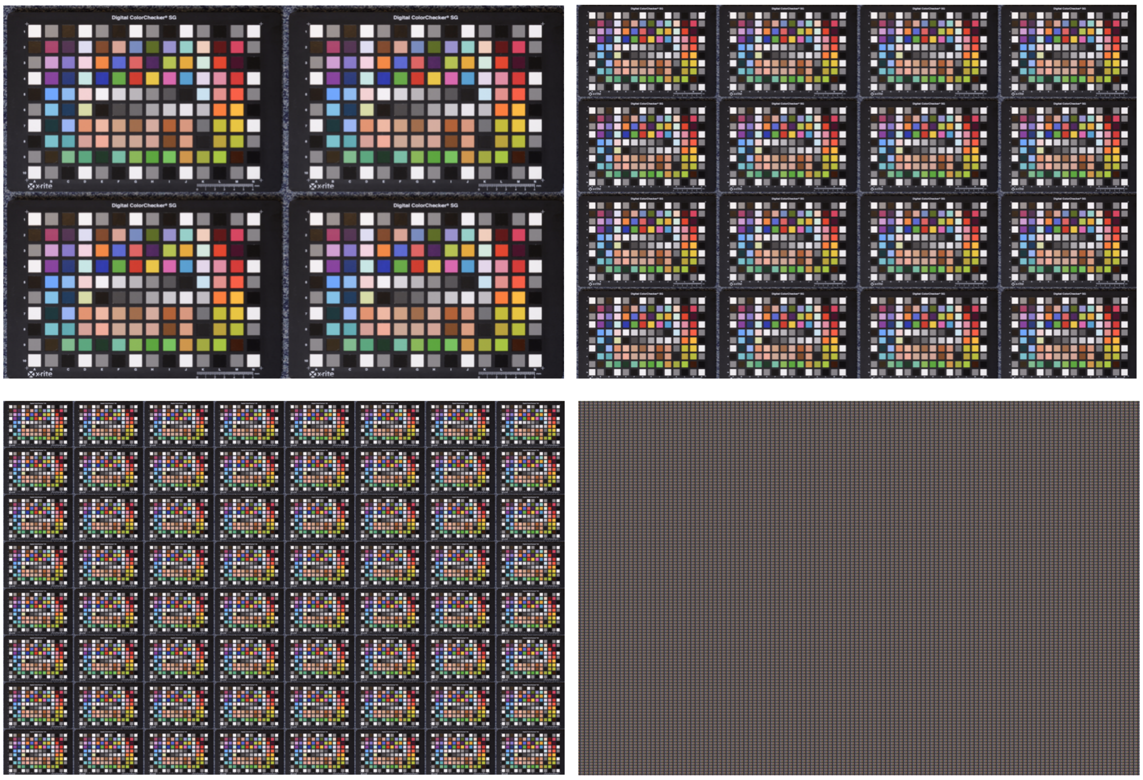


Fig. 5.3 As we replicate more and more images of the DC colour checker together, eventually all the colours blend into a uniform grey (bottom, right)

In Figure 5.3, we illustrate the blending of colours as we view a colourful object from *sufficiently far away*. Top left we show 4 copies the DC Macbeth Colorchecker. Then top right there are 16 copies and bottom left 64 copies and finally bottom-right 4096 copies. In this last example the individual colours in the colour chart can no longer be resolved. This is how a mosaic of DC Macbeth Colorchecker would appear from distance. Clearly, viewed from far enough away this colour target 'looks' grey.

As well as representing spectral data by its convex hull (more precisely, its convex closure), we also propose that the convex closure should be sampled uniformly and randomly. Of course, in doing so, we are changing the underlying distribution of the measured data. We argue that this is not a problem. Indeed, an argument can be made that we want to change the distribution. By uniformly and randomly sampling, we admit *all possible* physically plausible stimuli to be considered and every reflectance is equally as important as any other. In a sense, we are taking a sort of maximal ignorance stance (commonly used in colour imaging research [111, 112, 10]) and being agnostic – save for the fact that it is statistically plausible – of the likelihood of saying that one measured spectrum is more or less likely than another.

Given the assumption that we can represent a reflectance set by its convex hull (and that all reflectances within this hull, its convex closure, are possible) then this begs the question of how do we make use this representation. We will study this question in detail for the problem of colour correction.

Colour correction algorithms attempt to map camera RGBs to a human vision system referenced colour space, typically with a 3×3 matrix transform. RGBs could be mapped to the XYZ tristimulus values [131], to RGBs that drive a display (sRGB [86]).

The convex closure of reflectances is an infinite set of reflectances. This seems to pose a problem for regression-based colour correction. By utilising a single reflectance dataset, we can only carry out a regression based on the finite corresponding samples within this dataset. To sidestep this problem, we will show that the least-squares regression depends on the spectral autocorrelation matrix which, assuming 31 sampling wavelengths, is a 31×31 matrix. We may still use regression if we can solve for the autocorrelation corresponding to an infitite set of reflectances.

Unfortunately, for our convex closure of reflectances, calculating the autocorrelation is surprisingly laborious. Although it can practically be solved by random point generation, this brings us back to the question of "how many samples are required". To make the computation feasible, we represent reflectances by the enclosing hypercube of the data (which also includes the data's convex closure) described with respect to a privileged basis. This privileged basis is related to a PCA decomposition of the data. By using the privileged basis, the enclosing hypercube is a *tighter* fit than if the hypercube were applied in the spectral domain. This means by using the privileged basis, the hypercube is smaller and more tuned to the data. We can then show how we calculate the 3×3 colour correction matrix, in a least-squares sense – by integrating over the enclosing hypercube.

We make a couple of additional remarks. First, here and henceforth throughout this chapter we use hypercube to mean any cubic type object whose faces are orthogonal to each other (but they need not to have unit volume). Second, we note that an enclosing hypercube also contains the convex-closure. The convex closure integrates and generalises a dataset, and the hypercube generalises the convex closure even further.

Experiments establishs that integrating the reflectance sets (not sampling) leads to good colour correction when we train on a hypercube reflectance set and test on the individual sampled reflectances. Indeed, performance is similar to optimal least-squares using the sampled data alone. Significantly, when we train and test on different reflectance datasets (i.e. when we cross validate) our integration method continues to deliver good results that are competitive with the prior art. Finally, we show that, by representing reflectances by their hypercube enclosures, different reflectance datasets are much more similar to one another. This is an important result since the question of ‘what reflectance data should we use?’ is often posed in colour imaging research. The work we present here indicates the choice of which dataset is less important if it is integrated (i.e. all data sets become more similar to one another).

In section 5.2, we review colour correction and the idea of representing reflectances using a linear model. In section 5.3 we explain our core contribution solving for colour correction by integrating over a sampled data set. Experiments validate our method in section 5.4. The chapter concludes in section 5.5.

5.2 Representing Reflectance and Linear Least-Squaress Colour Correction

Let us begin by recapitulating the integrated response equation that relates the eye’s or camera’s response to a spectrum of light $E(\lambda)$ striking a surface with the spectral reflectance function $S(\lambda)$:

$$\underline{\rho} = \int_{\omega} E(\lambda) S(\lambda) \underline{Q}(\lambda) d\lambda \tag{5.1}$$

In Equation 5.1, $\underline{Q}(\lambda)$ is a vector function of 3 camera or three human vision system referenced sensors and $\underline{\rho}$ is the trichromatic response vector. The integral is computed over the visible spectrum ω, which runs from 400 to 700 Nanometres.

Let us consider some actual spectral measured data. Top left of Figure 5.4 we show the spectral reflectances for the top 6 patches of the Macbeth ColorChecker (For a picture of the Macbeth ColorChecker, please refer to Figure 1.1). Top right, we plot the spectral power distribution for a typical Daylight illumination. Finally, bottom, we plot the XYZ colour matching functions.

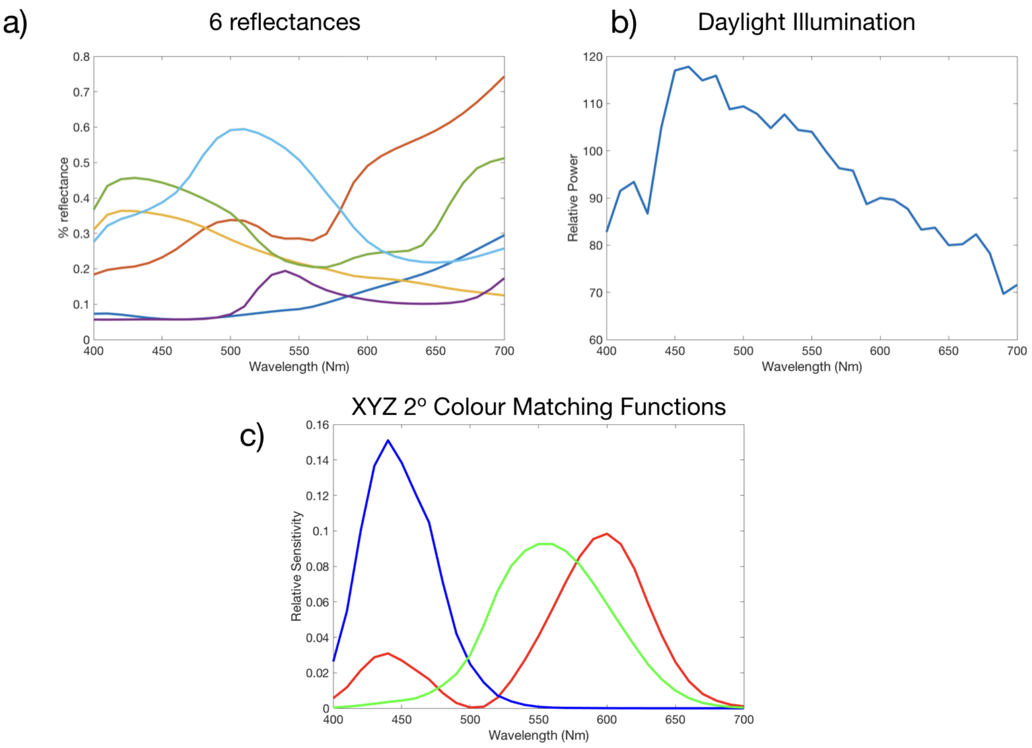


Fig. 5.4 In (a), (b) and (c) we plot spectral reflectance functions, a Daylight illumination and the XYZ colour matching functions

It is common to represent spectral quantities at 10 Nanometre sample points across the visible spectrum. This means that the spectral quantities $E(\lambda)$, $S(\lambda)$ and $Q_i(\lambda)$ ($i \in \{R, G, B\}$) in Equation 5.1 are represented by the corresponding 31-component vectors: $\underline{E}$, $\underline{S}$ and $\underline{Q}_i$ (where $\underline{Q}_i$ also incorporates the 10Nm sampling distance) . Grouping the 3 sensors in a 31×3 matrix Q, we can rewrite Equation 5.1 as:

$$\underline{\rho} = Q^{\mathsf{T}} diag(\underline{E})\underline{S}^{\mathsf{T}} \tag{5.2}$$

where $diag()$ makes a diagonal matrix from the vector argument and $^\intercal$ denotes vector/matrix transpose.

Let us represent a set of n surface reflectance spectra by the $31 \times n$ matrix S. Each row of the matrix represents the reflectances at a single wavelength, and each column is one reflectance spectrum. We will denote camera sensors and the XYZ colour matching functions (CMFs) by respectively R and Q (both 31×3 matrices). The camera and XYZ trichromatic responses to all the reflectances in S are computed is:

$$P = R^\intercal diag(\underline{E})S \tag{5.3a}$$

$$X = Q^\intercal diag(\underline{E})S \tag{5.3b}$$

Both P and X are $3 \times n$ matrices.

In Equation 5.3, we are calculating the XYZ response but we could also have calculated triplets for another human vision referenced colour space such as sRGB [86]. But, here and henceforth in this chapter we will calculate XYZs.

In colour correction, we seek to map RGBs to XYZs, typically with a 3×3 matrix. Unless the camera sensitivities are a linear transform from the XYZs – the so called Luther conditions [10] are met – the correction is inexact. The least-squares colour correction optimisation problem is written as:

$$\min_M ||MP - X||_F \ , \tag{5.4}$$

where $||\cdot||_F$ above denotes Frobenius norm.

Equation 5.4 can be solved in closed form using the Moore-Penrose inverse [132]:

$$M = XP^\intercal(PP^\intercal)^{-1}. \tag{5.5}$$

We now wish to consider the least-squares, Moore-Penrose, solution in more detail. To simplify matters it is useful to define a colour signal matrix as

$$C = diag(\underline{E})S \tag{5.6}$$

which means we can rewrite Equation 3 as:

$$P = R^\intercal C \tag{5.7a}$$

$$X = Q^\intercal C \tag{5.7b}$$

Let's substitute Equation 5.7 into Equation 5.5

$$M = Q^{\intercal}CC^{\intercal}R(R^{\intercal}CC^{\intercal}R)^{-1}. \tag{5.8}$$

Equation 5.8 teaches that the correction matrix M only depends on the 31×31 colour signal autocorrelation matrix $CC^{\intercal}$ and camera and XYZ spectral sensitivities.

We note, of course, that the least-squares solution is only one way to calculate the 3×3 correction matrix. Indeed, various authors [133, 134] suggest that we should find a matrix that minimizes a perceptual error. In other research the colour correction transform is not a matrix [104, 105]. This said the venerable linear matrix and least-squares is widely deployed not least because image formation itself is linear. Here we will use linear regression not only because it is tried and tested but also because computing the best least-squares transform for our enclosing hypercube of reflectances can also be computed (because we can calculate the autocorrelations for our infinite reflectance sets). It is not clear whether any other correction methods except least-squares regression can be easily computed (given the methods we develop in this chapter).

Note also that the development we have made so far tacitly assumes that we are correcting colours for one illuminant condition. In reality – especially for cameras used outside the lab – we have multiple lights. Assuming that we find the best 3×3 matrix for each light individually we can use the formulation set forth above.

However, sometimes we might find the single linear mapping that best maps the RGBs for all surfaces viewed under all lights to corresponding XYZs. Even for this circumstance the role of $CC^{\intercal}$ is important. Let us expand $CC^{\intercal}$ using Equation 5.6:

$$CC^{\intercal} = diag(\underline{E})SS^{\intercal}diag(\underline{E}) \tag{5.9}$$

In [135] it was shown that we could rewrite Equation 5.9 as:

$$diag(\underline{E})SS^{\intercal}diag(\underline{E}) = [\underline{E}\,\underline{E}^{\intercal}] \otimes [SS^{\intercal}] \tag{5.10}$$

where the operator $\otimes$ 'means' the component-wise multiplication of two matrices. Further in [135] it was shown that if there were m measured lights in the $31 \times m$ matrix E then the colour signal correlation matrix (all lights and all surfaces) equals:

$$CC^{\intercal} = [SS^{\intercal}] \otimes [EE^{\intercal}] \tag{5.11}$$

Again the central role of the spectral autocorrelation $SS^{\intercal}$ is clear.

Also, note that in the least-squares solution (Equation 5.5) both the autocorrelation and its inverse occur (in the least-squares fit formula). It follows that if we substitute $kCC^\intercal$ (k is an arbitrary scalar) that the least-squares fit matrix is unchanged.

$$M = Q^\intercal kCC^\intercal R(R^\intercal kCC^\intercal R)^{-1} = Q^\intercal CC^\intercal R(R^\intercal CC^\intercal R)^{-1} \tag{5.12}$$

From Equation 5.12, it both follows that the colour correction matrix is calculated is independent of scene exposure (a useful property) and second that we can divide the auto-correlation by a scalar (and this will facilitate calculating the similarity of two autocorrelations).

5.2.1 Representing Reflectance Data

In Figure 5.5, respectively, in panels (a), (b) and (c), we show the autocorrelation for the 170 object reflectances [96] , the 462 Munsell's [136] and the 120 Dupont reflectances [96].

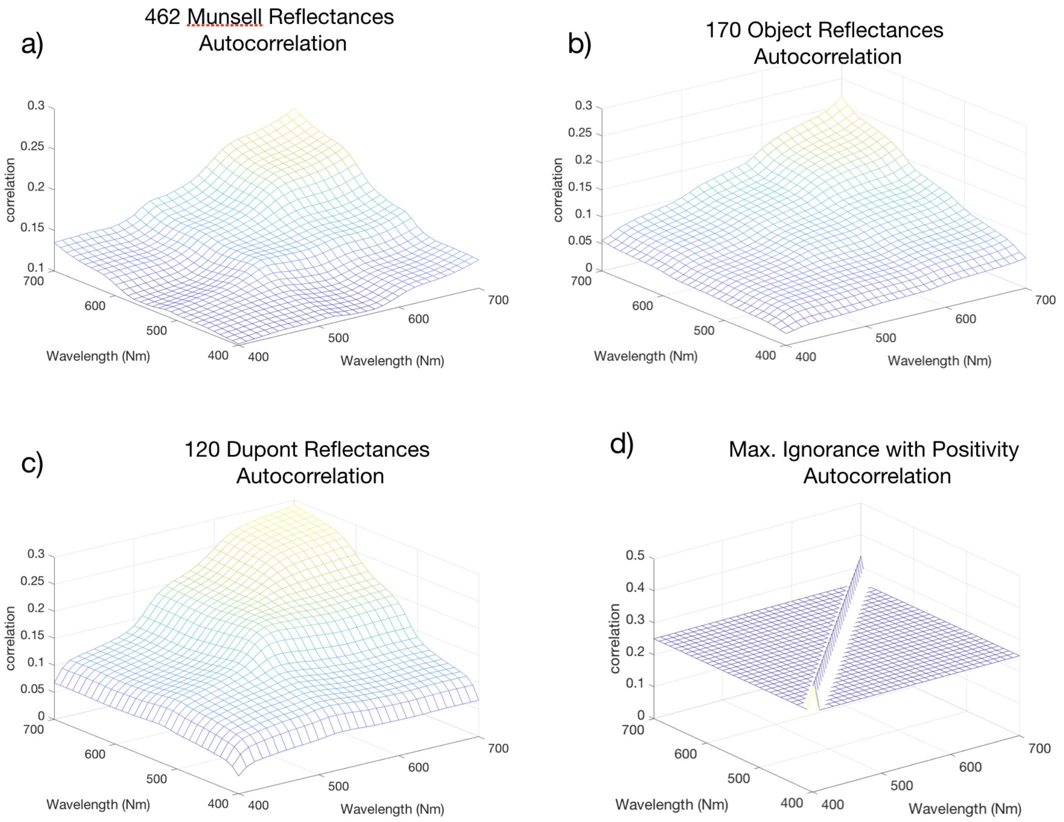


Fig. 5.5 in (a), (b) and (c) the autocorrelations, $SS^\intercal/n$ (n denotes the number of samples) of the Munsell, Object and Dupont reflectance datasets

Encouragingly, the autocorrelations look, broadly, similar. As a function of increasing wavelength each of the reflectance datasets appear more correlated. And, the further apart two wavelengths are, the less they are correlated. This said, the autocorrelations are quite different from one another. The different autocorrelations lead to significantly different colour correction matrices (see the reported experiments in Chapter 4 and more experimental presented later in this chapter). Looking at the shapes of the autocorrelations we might assume that the autocorrelation for the Munsells is the ‘outlier’. It is more different than the other too, and it appears the most ’bumpy’. Yet as we develop our method of integrating over reflectance sets we will, tantalisingly, discover that all data sets start to exhibit these bumps.

In Figure 5.5 panel d), we show the autocorrelation corresponding to the Maximum ignorance with positivity (MIP) assumption, where any vector that is all positive between 0 and 1 is equally likely. Relative to this assumption, very jaggy spectra as well as smooth spectra can occur even though the former do not appear in nature. We include the MIP autocorrelation to show how different it is from the autocorrelations of real reflectance data. Previous work has shown that the MIP assumption (i.e. adopting the MIP autocorrelation matrix) leads to poor colour correction performance.

Returning to Figure 5.4, we see that the 6 Macbeth ColorChecker reflectance curves plotted are smooth. Smoothness, at least in first-order, is a property of all naturally occurring reflectance. Because surfaces are smooth they can be represented by an m-dimensional linear model (where $m << 31$):

$$S(\lambda) \approx \sum_{i=1}^{m} U_i(\lambda)\sigma_i \;\equiv\; \underline{S} \approx U\underline{\sigma} \tag{5.13}$$

here U_i (or $U_i(\lambda)$) is called a basis function and σ_i is a scalar weighting its contribution, $\underline{\sigma}$ is a $m \times 1$ vector. In Equation 5.13, U is a $31 \times m$ matrix and represents the discrete linear basis for modelling reflectances. The interesting question, of course, is what is the value of m. Different studies have concluded that the answer is between 6 and 9 basis functions [124, 137]. Though, for some applications even a 3-dimensional basis set suffices [138, 62].

For an n-reflectance data-set S and a fixed dimension m we would like to find the basis U that best approximates:

$$min_{U,\Omega}\; ||S - U\Omega||^2 \tag{5.14}$$

where U and Ω are respectively a $31 \times m$ and $m \times n$ and the ith column of Ω is $\underline{\sigma}_i$. Characteristic Vector Analysis (like PCA where the mean is not subtracted) is the apposite tool. CVA returns the optimal solution to Equation 5.14.

The CVA solution has a number of attractive properties. First, the basis matrix U is orthonormal, $U^{\intercal}U = I_{m\times m}$. Second, the basis is ordered. The first basis function is the direction (among all others) that best captures the variance in the spectral data S. The second basis function is the direction orthogonal to the first that captures the most variance. If U^m denotes the basis for m dimensions then $[U^m\ \underline{u}^{m+1}]$ denotes the solution for $m+1$ dimensions (where $[\underline{u}^{m+1}]^{\intercal}U^m = \underline{0}$, $\underline{0}$ is the m-component vector of 0s).

Further, in CVA, the $m\times m$ matrix $\Omega\Omega^{\intercal}$ is the diagonal matrix D^2 with all positive diagonal components that decrease monotonically, from first to last diagonal component. Placing the diagonal components of D^2 in the vector $\underline{d}$, the ratio $(\sum_{i=1}^{k} d_i)/(\sum_{i=1}^{m} d_i))$ reports the % of variance captured by the first k basis functions.

It follows we can write Ω as $DV^{\intercal}$ (note D is the diagonal matrix that is the square-root of D^2) where V is also orthonormal. Effectively, we recapitulated – what is in fact the evident truth for those that have seen similar equations before – the Singular Value Decomposition of reflectances.

$$S \approx UDV^{\intercal} \tag{5.15}$$

From which it follows that we can decompose the autocorrelation of $SS^{\intercal}$ as:

$$SS^{\intercal} \approx UD^2U^{\intercal} \tag{5.16}$$

Again, the approximation in 5.16 is also that best least-squares approximation: the SVD decomposition of the autocorrelation suffices to find the optimal basis U. Once more, we see that what is important in datasets is not the reflectance set *per se* but the autocorrelation (from which the optimal basis can be found). See [118] for a wider review of representing spectra by characteristic vector analysis.

In Figure 5.6, we respectively show the best 3-dimensional basis for the Munsell, Object and Dupont Reflectances. These 3-dimensional basis sets respectively capture 99.63%, 99.14% and 99.48% of the variance in the dataset. Unsurprisingly, as for the autocorrelations we find the ‘trend’ in the data to be similar: the basis functions appear similar but there are significant differences in their shapes.

The basis functions (or coordinate axes) are ordered in terms of importance – actually in terms of the amount of variance in the dataset that falls in the direction of the basis function – and this is colour coded in Figure 5.6. The 1st characteristic vectors are shown in blue, the second in red and the third in orange. Notice that these curves increase in complexity (they become less straight). The blue lines are almost straight (DC terms). The red curves are smooth and cross the x-axis once (abstractly, they resemble a function like *cos* $0.5x$ (where the period of x is the visible spectrum). The 3rd characteristic vectors – again very abstractly looks like $-cos\,x$.

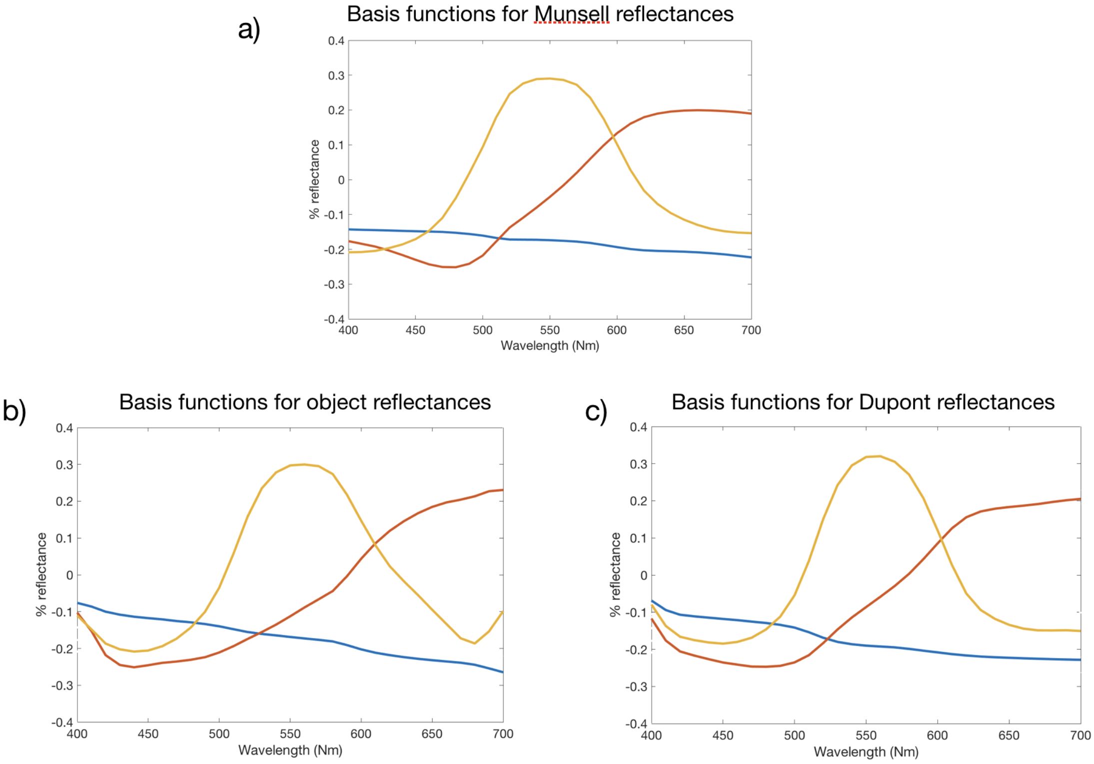


Fig. 5.6 in (a), (b) and (c) the best 3-dimensional bases for the Munsell, Object and Dupont reflectance datasets

With respect to each basis lets reconstruct, optimally in a least-squares sense, a reddish reflectance spectrum, from the checker in Fig. 5.4. The result is shown in Figure 5.7. Visually, none of the ‘fits’ seems that good: they all have a significant error compared to the original.

If we measure the error in spectral recovery as $||\underline{S} - \hat{\underline{S}}||/||\underline{S}||$ – where $\hat{\underline{S}}$ denotes the closest spectrum in the span of one of our 3-dimensional basis sets – then respectively, the 3 reconstructed spectra have reconstruction errors of 13% (Munsell basis), 13% (Object basis) and 9% (Dupont basis). Indeed, it is well known in the field that the high % variance captured method (>99%) is an overly optimistic measure of how well a basis set models individual reflectance spectra. Significantly, the percentage error between the recovered spectra themselves is also high e.g. the % error of the recovered estimate using the object basis compared with the Munsell recovery is 9%.

In Equation 5.14, we recapitulated how the singular value decomposition is used to find the best m dimensional basis U for a given reflectance data set. We are interested in finding coordinates in the basis that best approximates the ith reflectance in a dataset:

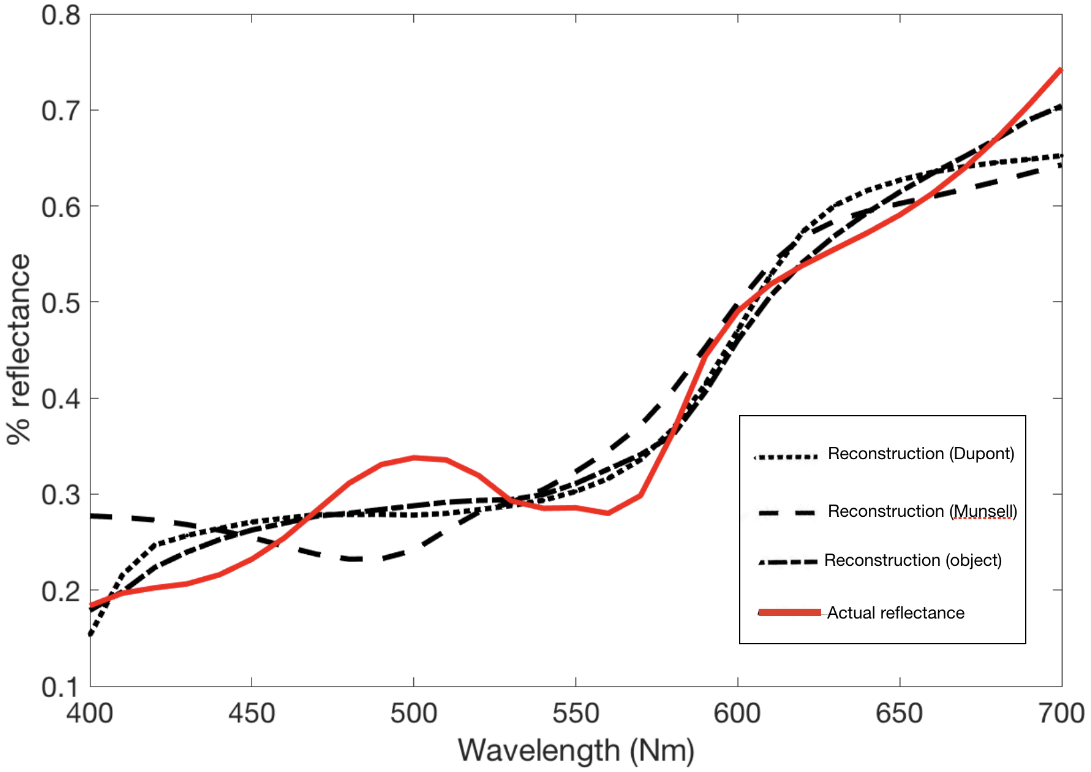


Fig. 5.7 Red, is the actual reflectance. Three 3-dimensional fits are shown (using the bases derived from the Munsells, Object and Dupont datasets)

$$\underline{S}_i = U\underline{\sigma}_i \tag{5.17}$$

We find the $\underline{\sigma}_i$ that minimizes $||\underline{S}_i - U\underline{\sigma}_i||$ via the Moore-Penrose inverse

$$\underline{\sigma}_i = [U^{\mathsf{T}}U]^{-1}U^{\mathsf{T}}\underline{S}_i \tag{5.18}$$

Since U is orthonormal,

$$\underline{\sigma}_i = U^{\mathsf{T}}\underline{S}_i \tag{5.19}$$

and

$$U^{\mathsf{T}}S = \Omega \tag{5.20}$$

Let's plot, in 3 dimensions, the coordinates of the 120 Dupont reflectances relative to the optimal basis (itself plotted in Figure 5.6).

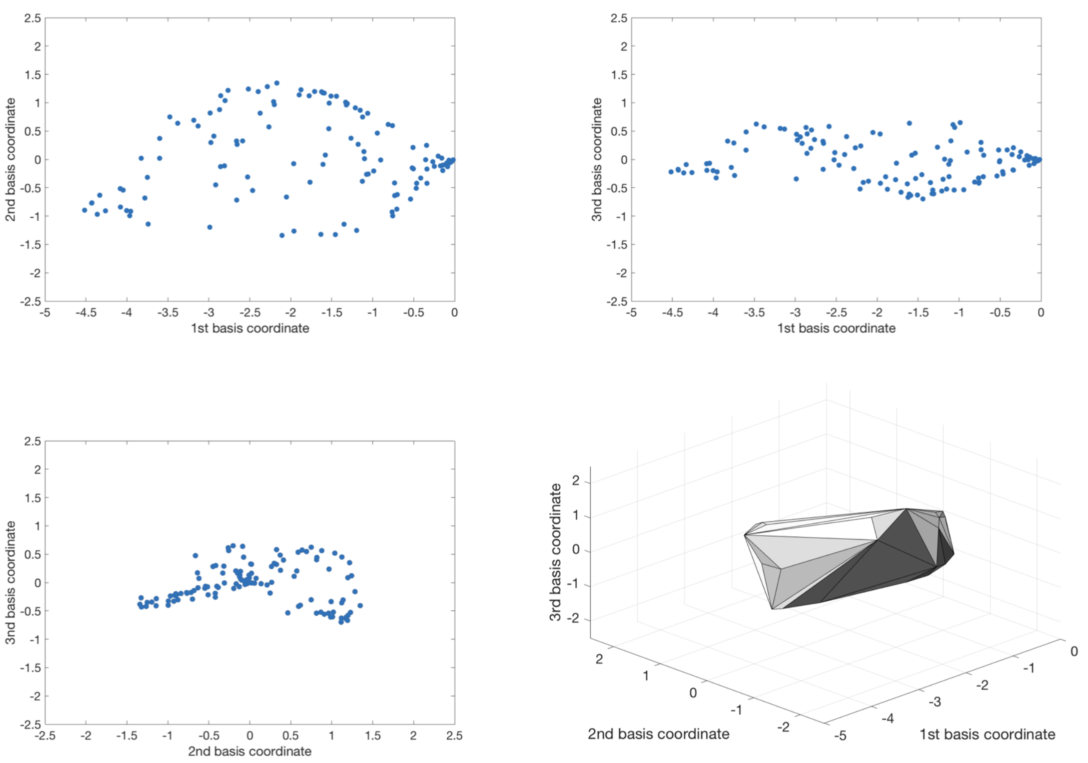


Fig. 5.8 For a 3-dimensional basis we plot the coordinates for the Dupont reflectance set (each coordinate against the other two). Bottom right the plot of the 3-dimensional convex hull

In Figure 5.8, top left, we plot the 1st against the 2nd basis coordinates. Top-right and bottom-left we plot 1st against 3rd and 2nd against 3rd. Bottom-right we plot the convex hull of the coordinates. Looking at the figure we see that the range of coordinates diminishes as the dimension number increases. For this set respectively, the 1st, second and 3rd coordinates lie in the intervals [0,-5], [-1.25,1.25] and [-.625,.625]. That is the spread of the data (roughly) halves per dimension.

5.3 Integrated Reflectance Sets

As presented the last section, least-squares regression colour correction is driven by the autocorrelation of the reflectance data set $SS^{\intercal}$ (or the autocorrelation of the colour signal matrix that itself also depends on the reflectance autocorrelation). Figure 5.5, shows that the auto-correlation varies for different reflectance datasets. Moreover, even the best basis – the best coordinate system for describing reflectances – also varies with reflectance data set, see Figure 5.6 and Figure 5.7.

In this section we make 3 contributions. First, we re-describe all reflectance datasets with a common basis and we discuss how such a basis is computed. Second, we develop our convex closure representation: i.e. the idea that the convex hull of a reflectance set delimits all physically plausible reflectances. For applications like colour correction we need to integrate over the convex closure e.g. to calculate the best colour correction transform. But, integrating over convex sets is hard. Thus our third contribution, is directed towards making computations feasible. We represent reflectance data by the enclosing hypercube with respect to the common basis coordinates.

5.3.1 Deriving a common reflectance basis

As illustrated in Figure 5.6, different reflectance data sets have different basis functions. This fact alone means it is difficult – if we use a low dimensional representation of reflectance – to compare one reflectance set against another. We can compare basis coordinates (the $\underline{\sigma}$) if and only if we use the same basis functions. To solve this problem we will find a *single* set of basis functions which is suitable to represent *all* reflectance datasets.

The Statistical Approach: One might start by combining the autocorrelation of the different datasets using a weighted sum:

$$SS^{\mathsf{T}} = \sum_i \frac{S_i S_i^{\mathsf{T}}}{n_i} \tag{5.21}$$

where S_i is the set of reflectances in dataset i and n_i is the number of samples in this dataset. This weighted summation gives each reflectance dataset equal weight on its influence on the final basis functions, regardless the size of the reflectance dataset itself. Now, as described in the last section, single value decomposition (SVD) [139] is used to find the optimal $31 \times m$ basis matrix U.

The Cosine Series expansion. As commented previously (see discussion of Figure 5.6), the optimal basis functions (set of characteristic vectors) found for a reflectance set – at least from an abstract vantage point – looks somewhat like the first three terms in discrete cosine series expansion. Thus, it is natural that we consider using the discrete cosine expansion as the common basis. Importantly, a discrete cosine basis has well known (optimal) energy compaction properties [140]. It is a reasonable choice for the common basis if we wish to be a priori agnostic about the shape of reflectances.

As a cosine series, the basis matrix U is written in closed form. Here $i \in \{0,1,2,\cdots,31\}$ and $j \in \{1,2,\cdots,m\}$

$$U_{ij} = \cos\left(\frac{(j-1)\pi}{N}\left(i+\frac{1}{2}\right)\right)/k_j. \tag{5.22}$$

where the form of above equation is chosen so that U is orthogonal. The scalar k_j chosen so that the individual columns of U have magnitude 1 (it accounts for the Wavelength sampling).

Henceforth, we will use cosine series basis in the following discussion and in the experimental section.

5.3.2 Convex- and Hyper-cube closures of reflectances

Physically, if we have a checker-board pattern comprising two reflectances $S_1(\lambda)$ and $S_2(\lambda)$ in equal proportions then if we view this checker board from far enough away the checker will appear to have a single colour (with an effective reflectance $0.5S_1(\lambda)+0.5S_2(\lambda)$). More generally, viewed at distance a texture of colours blends to a single colour (with the new effective reflectance being a convex sum of the individual reflectances in proportion to their $\%$ area coverage). That is, given a dataset of reflectances, if they are *allowed* appear in patterns of arbitrary shape and complexity, then all convex combinations of the reflectances in a dataset might be physically measured.

It follows that the set of plausible spectra is the convex closure of the reflectance samples found in a given data set. And, this convex closure in turn is delimited by the convex hull of the dataset. Unfortunately, calculating a convex hull in d dimensions (for reflectances d is 31) is computationally hard (complexity $O(n^{\lfloor \frac{d}{2} \rfloor})$, [141]). Even for 8 dimensional data and a few 100s of reflectances the problem is intractable (at least given the current implementation in Matlab). In the 8-dimensional case, given 1000 points leads to on the order of 1 trillion operations to solve for the convex hull!

Suppose we can compute the convex hull of a reflectance set. For colour correction we need to integrate over this set. Arguably, the cost of integrating over a convex hull is even greater than the cost of computing the hull itself. Perhaps the simplest integration problem is computing the volume of the convex hull. Here, the cost of computing an analytical solution is the same as the convex hull computation. Indeed, the cost is sufficiently high, that volumes are generally computed via a Monte Carlo simulation.

To illustrate the Monte Carlo simulation suppose we are given a convex hull and we compute the bounding hypercube (the min and max coordinates in the hull per dimension, an easy computation). The volume of this enclosing hypercube is the product of the dimensions (of the hypercube). For the purposes of this example, suppose the volume of the hypercube is k. Now, we generate p points uniformly and randomly in the cube. For each point we check if it is inside or outside of the convex hull (a fast operation). If we find that 90% of the points are inside the hull then the computed volume – by random sampling – is $0.9k$. Of course, we have to choose *enough* random points to calculate the volume with good accuracy. From the introduction to thos chapter 'enough' can be very large.

To sidestep the issue of the complexity of integration, we will find the enclosing hypercube of the data with respect to the common basis. This turns out to be a simple computation (the complexity is $O(d)$), as we discussed above in our volume computation example. For a d dimensional dataset S the coordinates of S in basis U are equal to $\Omega = U^{\intercal}S$ (see Equation 5.20). Denoting the kth row of Ω as Ω_k, the min and max values (m_k and M_k) for the coefficient σ^k are calculated:

$$m_k = \min(\Omega_k) < \sigma_k < \max(\Omega_k) = M_k \tag{5.23}$$

The $2d$ min and max coordinates in $\underline{m}$ and $\underline{M}$ delimit a hypercube. We illustrate this idea by redrawing Figure 5.8 making clear the bounding hyper-cube (in this case the 3 dimensional rectangle plotted in red in Figure 5.9).

Notice there is a significant 'gap' between the boundary of the points and the edge of the cube. In fact, for this example, the enclosing hypercube has more than twice the volume compared to the convex set.

The physical meaning of the 'gap' is that under the hypercube model we admit reflectances that we both haven't seen before and may well be non-physically realizable. That is, they have either reflectivity less than 0 or greater than 1. This kind of extrapolation is common. The Maximum Ignorance, Maximum Ignorance with positivity and Toeplitz assumptions [10, 112, 142] all effectively admit non-physically realizable reflectances.

Under the Maximum ignorance with positivity assumption, we allow each spectral wavelength to be in the interval [0,1] and the value selected per wavelength to be independently of all other wavelengths. Here, all reflectances in the 31-dimensional unit cube are equally likely. For our worked example, we can also compute the bounding hypercube – relative to the Dupont 3-dimensional basis – for the Maximum ignorance assumption with positivity. The corresponding MIP hypercube is shown in blue. The MIP cube is 8 times than the red hyper-cube based on real data (so, a substantially weaker assumption that admits many more infeasible reflectances).

The importance of the coordinate space

The reader might wonder why all the analysis building to this point has represented spectra with respect to a basis. The reason is that we are seeking a representation where the bounding

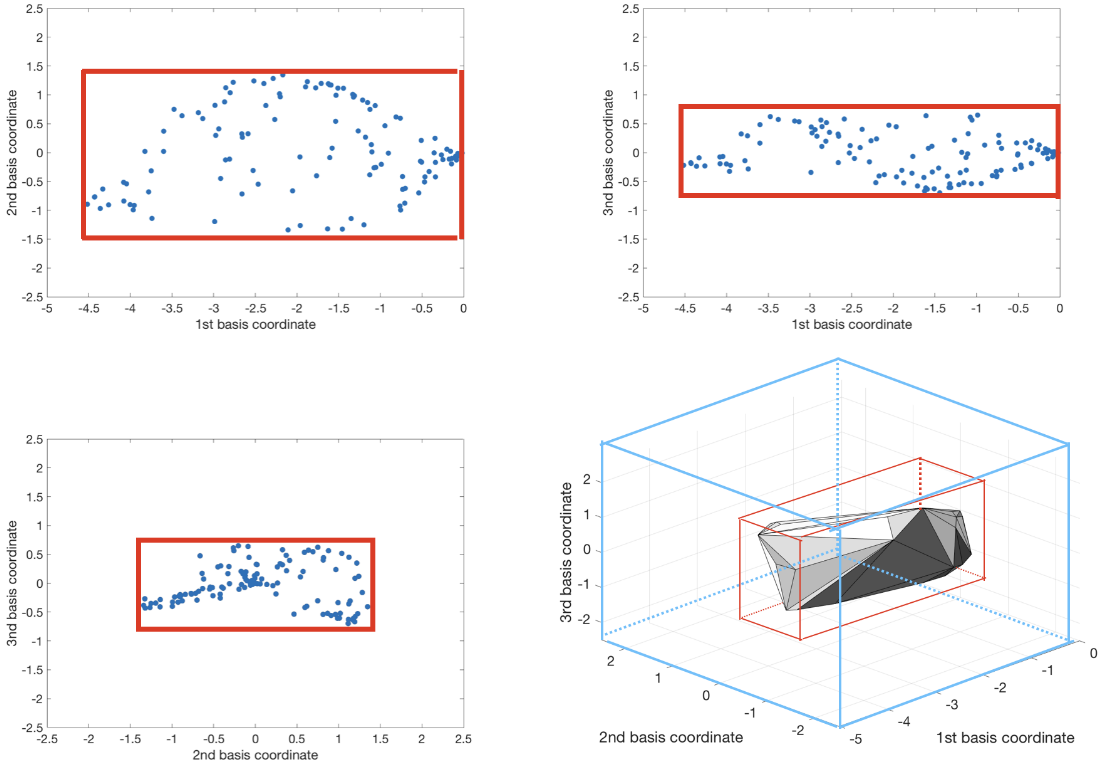


Fig. 5.9 For the Dupont data set we plot the coordinates with respect to an optimal basis. The enclosing hypercube is shown in red. Assuming we projected all possible spectral data (the maximum ignorance with positivity assumption) on the same basis the corresponding enclosing hypercube is shown in blue.

hypercube idea makes sense. While we are willing to allow some spectra that are not in the convex closure of the data set, we do not wish to admit too many.

Arguably, by decorrelating the data, CVA – or indeed the cosine basis that achieves a similar decomposition – finds a representation that is more cube-like. Indeed, data uniformly randomly distributed in a cube is completely decorrelated (knowledge of one coordinate tells you nothing about the other). And, of course by coding a reflectance with respect to a basis, we can choose to use fewer than 31 parameters (fewer than the total number of wavelengths).

Suppose instead we carried out a similar analysis in the primal domain (the primal domain here is the reflectance spectra themselves). For each wavelength, we can calculate the max and minimum reflectance values. In all likelihood this will give reflectances close to 0 and 1 respectively since there are very dark and very bright colours of all hues. If, per wavelength, we assume that reflectance lies in [0,1] and that one wavelength is not correlated with another then this is the definition of the Maximum ignorance with positivity assumption.

This assumption leads us to accept many reflectances (think very jaggy spectra) which are not similar to any actual measured data.

We note that the volume of a geometric object does not change when we rotate the object (or more generally apply a unitary transform). When we project reflectances onto a basis, such as the cosine expansion we are re-describing the data with respect to new axes but we are not changing the shape of the data. As a test we projected the 120 Dupont data onto the full 31-dimensional cosine expansion (i.e. we applied a full rank unitary 31x31 matrix) to the reflectances. Using the bounded hypercube we can calculate the volume before and after the unitary transform. Literally, we find the volume with respect to the discrete cosine basis to be orders of magnitude less than in the primal wavelength domain. That is the minimum bounding hypercube depends on the basis in which data is described.

We illustrate this concept in Figure 5.10 for the data shown (blue dots). in 2 dimensions. The rectangle at 45 degrees has a large bounding triangle (according to the primal axis). Rotating to a new basis returns a bounding box that more closely describes the shape of the underlying data.

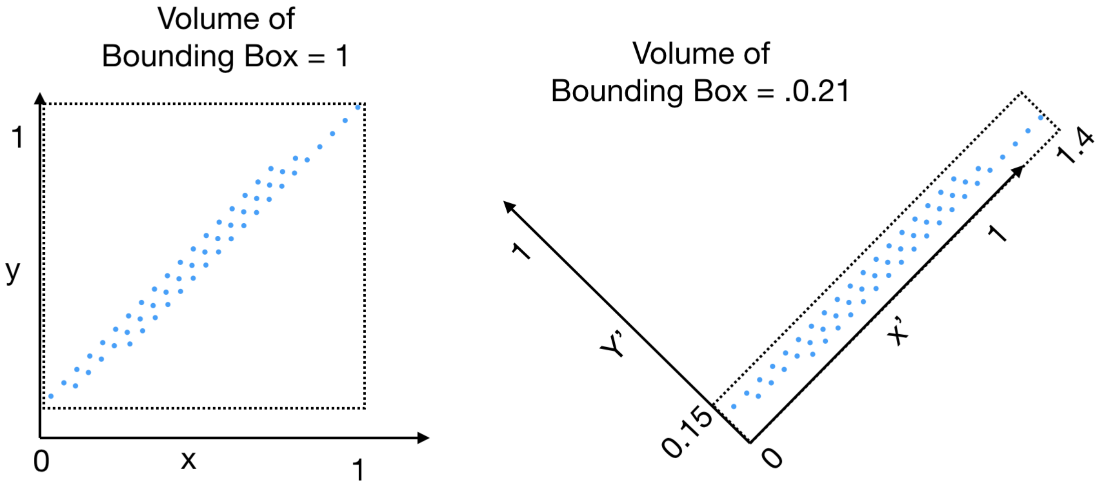


Fig. 5.10 In the left we see points scattered at 45 degrees. Relative to the x- and y- axes the bounding box – enclosing hypercube – of the data has area 1. Right, we rotate the axes 45 degrees (analogous to CVA). Now the bounding box fits the data better and has area 0.21.

In our method, it is important to compute the bounding hypercube of the basis coordinates and not of the reflectances themselves.

5.4 Calculating the autocorrelation matrix from the Reflectance Hypercube

We have already seen that the least-squares computation is carried out given knowledge of the spectral sensitivities of the camera and XYZ sensors, the autocorrelation and the light (or lights). In the *discrete* world the autocorrelation is simple to compute:

$$auto(S) = \frac{SS^{\mathsf{T}}}{n} \tag{5.24}$$

where n is the number of surfaces in the reflectance dataset. We would like to calculate the autocorrelation given the bounding hypercube. First, remember that in our basis representation we can write $S = U\Omega$ where U is the $31 \times m$ reflectance basis and each column of (the $m \times n$ matrix) Ω is the basis coordinates that define a single reflectance. Here, and henceforth – unless otherwise stated – we will assume $m = 31$ and that U is the cosine series basis. Moreover, we will set $m = 31$ so that U plays a decorrelating role – allows a bounding hypercube to better fit the data – but the transform is lossless. It follows that

$$auto(S) = \frac{U\Omega\Omega^{\mathsf{T}}U^{\mathsf{T}}}{n} \tag{5.25}$$

Given that U is fixed, the autocorrelation for a reflectance set depends on $\Omega\Omega^{\mathsf{T}}$ (the autocorrelation of the coordinates of the reflectances with respect to the basis).

$$auto(\Omega) = \frac{\Omega\Omega^{\mathsf{T}}}{n} \tag{5.26}$$

Let us rewrite Equation 5.26 as a summation:

$$[auto(\Omega)]_{ij} = \sum_{k=1}^{31} \Omega_{ik}\Omega_{jk} \tag{5.27}$$

for the special case when $i = j$,

$$[auto(\Omega)]_{ii} = \sum_{k=1}^{31} \Omega_{ik}^2 \tag{5.28}$$

When we come to computing the autocorrelation over the hypercube, the special case of Equation 5.28 is important. In Equation 5.27 (when $i \neq j$) there are two statistical variables but in 5.28 there is only one. This an important detail.

Let us consider for a moment three dimensional data bounded by the min and max values in each of the 3 dimensions. In the 3-dimensional case we have 3 minimum coordinates – in the x-, y- and z-directions – and 3 maxima. If we denote the coordinate of a vertex on the cube as (x, y, z) then independent of other dimensions $x \in \{m_x, M_x\}$, $y \in \{m_y, M_y\}$ and $z \in \{m_z, M_z\}$; that is by selecting all combinations there are $2^3 = 8$ combinations (the 8 vertices of the hypercube). In n dimensions we have n minima and n maxima, 2^n combinations i.e. there are 2^n vertices on an n dimensional hypercube,

Remembering that the ith row of Ω is denote Ω_i, and that the hypercube bounding Ω is defined by the min and max row coordinates in Ω.

$$box(\Omega) = \{\underline{m}, \underline{M}\} \text{ , where } m_i = min(\Omega_i) \text{ and } M_i = max(\Omega_i) \tag{5.29}$$

We denote the autocorrelation of all the reflectances in a bounding hypercube for Ω as $auto(Box(\Omega))$. To calculate $auto(Box(\Omega))$ we need to integrate over the bounding box.

5.4.1 Discrete versus continuous estimation

Before showing how we integrate over a hypercube it is informative to review how we calculate expectations in the discrete versus continuous domains. As an example, let us suppose that points belong to a uniform distribution in the interval [3,5]. Now 10 points are drawn uniformly and randomly sampled from this distribution. We find the expected value of the sample set by summing up the 10 points and dividing by 10. As, defined – points belong to a uniform distribution in the interval [3,4] – we know that the average should be 4. But, because of there being only small numbers of samples the average will likely be a little bit less or greater than 4.

Denoting each discrete sample as s_i we can write the expected value $E(s)$

$$E(s) = \frac{\sum_{i=1}^{10} s_i}{10} \tag{5.30}$$

Given we know that the underlying distribution is uniform, we can in fact find the expected distribution analytically i.e. without sampling. Rather we can sample every point (an infinity of points) in the interval and calculate the average. The apposite tool for this calculation is integration. In the continuous case we write

$$E(s) = \frac{\int_3^5 x\, ds}{\int_3^5 1\, ds} = \frac{\frac{x^2}{2}|_3^5}{x|_3^5} = \frac{8}{2} = 4 \tag{5.31}$$

Usefully, here we get exactly the correct answer (since we have sampled all points). Note that the integral and discrete formulations are similar. Indeed $\int_3^5 x\, ds$ means summing up x over the interval $[3, 5]$ and this is analogous to the summation in Equation 5.30. The denominator terms of Equation 5.30 is analogous to the sample size. We must divide by the interval length (analogous to dividing by the number of samples).

Suppose, as it will be relevant to integrating over hypercubes, that we have two samples s and t defined over intervals $[a,b]$ and $[c,d]$. The expected value of the product st in the continuous domain is calculated as:

$$E(s) = \frac{\int_a^b \int_c^d st\ dt\ ds}{\int_a^b \int_c^d 1\ ds} = \frac{\int_a^b \int_c^d st\ dt\ ds}{(d-c)(b-a)} \tag{5.32}$$

Here $1/((d-c)(b-a))$ is the area over which we are taking the double integral.

5.4.2 Integrating over a hypercube

Because the hypercube is a continuous set, the summations in Equation 5.29 become integrals. We will use the notion σ_i to denote a sample in the ith coordinate (σ_i denotes a continuous variable). Given that the enclosing hypercube is defined by the min and max coordinates in each dimension (see Equation 5.26) then $m_i < \sigma_i < M_i$. The Expected value of the autocorrelation is computed as:

$$[E(auto(Box(\Omega)))]_{ij} = \begin{cases} \frac{\int_{m_i}^{M_i} \sigma_i^2\ \mathrm{d}\sigma_i}{M_i - m_i} & \text{when } i = j \\ \\ \frac{\int_{m_j}^{M_j} \int_{M_i}^{m_i} \sigma_i \sigma_j\ \mathrm{d}\sigma_i \mathrm{d}\sigma_j}{(M_i - m_i)(M_j - m_j)} & \text{when } i \neq j. \end{cases} \tag{5.33}$$

We have different expressions for the diagonal and off diagonal terms for the same reason as the discrete case (see Equation 5.27 and Equation 5.28 and discussion). For the diagonal case there is only one statistical variable but for the off diagonal there are two. This in turn means computing the expectation along the diagonal of the autocorrelation is a single integral problem and the off diagonal computation involves solving a double integral.

Notice that the numerator terms of the continuous integral Equation 5.33 look similar, as we would expect, to the discrete summations. The denominator terms are different from ‘dividing’ by n (of Equation 5.27 and Equation 5.28). When we compute expectations in the continuous domain, as commented above, we need to divide by the length of the interval (for the 1-d case) and the area of integration for the 2-D case. Equation 5.33 can be solved for in closed form with the autocorrelation defined as:

$$[E(auto(Box(\Omega)))]_{ij} = \begin{cases} \frac{M_i^3 - m_i^3}{3(M_i - m_i)} & \text{when } i = j \\ \\ \frac{M_i^2 M_j^2 + m_i^2 m_j^2 - m^2 i M_j^2 - M_i^2 m_j^2}{4(M_i - m_i)(M_j - m_j)} & \text{when } i \neq j. \end{cases} \tag{5.34}$$

For the continuous case (integrating over the bounding hypercube) the estimated autocorrelation is calculated as $UE(auto(Box(\Omega)))U^{\intercal}$.

5.4.3 Integrating over a smaller hypercube

Suppose we calculate a scaled hypercube that is k times the size in each dimension

$$sBox(\Omega, k) = \{\underline{m}, \underline{M}\} \text{ , where } m_i = k \cdot min(\Omega_i) \text{ and } M_i = k \cdot max(\Omega_i) \tag{5.35}$$

From Equation 5.34, it is easy to show that:

$$E(auto(sBox(\Omega, k))) = k^2 E(auto(Box(\Omega))) \tag{5.36}$$

That is, by scaling the bounding coordinates of the hypercube the corresponding autocorrelation also scales. From Equation 5.12 we know that least-squares colour correction is independent of the magnitude of the autocorrelation. Moreover, and perhaps more importantly, we have shown that the terms in the autocorrelation scale to the square of the scaling applied to the data.

5.5 Experiments

In this section, we will review the sampled reflectance datasets we are going to use in our experiments. We then go on to consider how similar or different the sets are. After all, researchers often ask "what reflectance dataset should I use to validate my algorithms?". One prediction of our enclosing hypercube representation of reflectances is that post-enclosure different reflectance sets should be more similar to one another. For the purposes of this chapter – which is using colour correction as an exemplar application – we measure the similarity of reflectance sets by comparing the similarity of their autocorrelations. We also view the similarity question through the lens of information theory and ask how expensive it is to code one distribution given the optimal code book of another. Finally, the practical import of our work is investigated in colour correction experiments. Importantly, the goal here is not to beat least-squares (least-squares is least-squares optimal). Rather, a good result would be if training on enclosing hypercube data led to competitive performance.

There are 9 Tables in this section. These correspond, broadly, to 3 classes of experiment (autocorrelation similarity, bit-encoding and colour correction). In Figure 5.11 we provide a quick crib sheet that describes how to interpret the numbers in the tables (we describe the tables where they appear in the text but this reference may help in comparing the trends in the data).

5.5.1 Reflectance Datasets

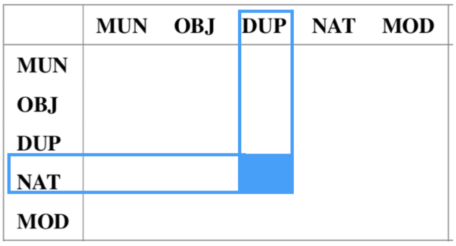
Hitherto in this chapter, we have considered the respectively 462, 170 and 120 reflectance **MUN**sell, **OBJ**ect and **DUP**ont reflectance sets. The Munsell reflectances are painted patches designed to have a large colour gamut. The Object dataset contains spectral of typical objects including bricks, wood and pavement. The Dupont set contains the spectra of colourful dyed material. All of these datasets are similar in that they do not a priori place constraints on the shape of spectra.

To these three exemplar measured reflectance datasets, we add two more. First is the **NAT**ural dataset measured by Westland *et al* [92] which comprises 404 measured spectra of plants, foliage and flowers. Second, we add a **MOD**el that comprises 500 randomly selected spectra from the bounding hypercube of the bounding hypercubes of the **MUN**, **OBJ**, **DUP** and **NAT** datasets.

From the discussion in chapters 2 to 4, there are other natural relfectances we could have used. We do not report results for those sets here because the trend in the data and the conclusions we draw are the same. Indeed, we are conscious that we are, in this chapter, presenting a large amount of experimental data. So, a priori, we are choosing to convey the *necessary* set.

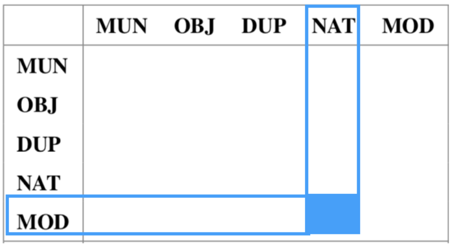
We do, however, actively reject the manufactured colour targets (e.g. Agfa IT8) because they are printed. Consequently, every patch is a mix of 4 inks (CMYK) and all have a characteristic smooth shape and all are naturally low-dimensional. We purposely do not include experimental results for the printed datasets here (though the methods we have developed do work for these colour targets).

Let us review the construction of the **MOD** dataset. Using the index k, $k \in \{\mathbf{MUN}, \mathbf{OBJ}, \mathbf{DUP}, \mathbf{NAT}\}$, we calculate the enclosing hypercube – denoted by ENC – of these sets (where as before the spectra are represented by their Ω coordinates with respect to the Cosine Series Basis U):



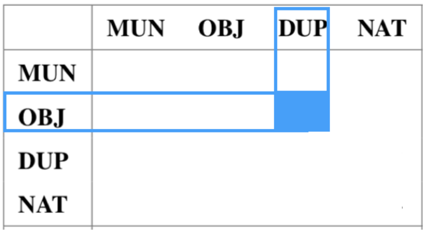
Here, we calculate the % similarity w.g. ||**NAT**-**DUP**||/||**NAT**||

Tables 1 and 2 report
% autocorrelation similarity



Here, we calculate the KLD where we code **NAT** with the codebook for **MOD**

Tables 4 and 5 report
Kullback-Leibler Divergence



Here, we calculate 3x3 matrix that best takes RGBs for the **OBJ** dataset to corresponding XYZs. We then use this matrix to colour correct **DUP** RGBs to corresponding XYZs

Tables 6 to 9 report
Delta E colour correction error

Fig. 5.11 Crib Sheet: Top, how to interpret the autocorrelation experimental tables. Middle teaches the interpretation of Kullback Leibler divergence (viewing similarity by bit counting). Bottom we descive the data in the Colour Correction Tables.

$$\begin{aligned} box(\Omega_{\mathbf{ENC}}) &= \{\underline{m}_{\mathbf{ENC}}, \underline{M}_{\mathbf{ENC}}\} \\ \underline{m}_{\mathbf{ENC}} &= \min_k \underline{m}_k \\ \underline{M}_{\mathbf{ENC}} &= \max_k \underline{M}_k \end{aligned} \tag{5.37}$$

For the ith of 500 trials we select a vector $\underline{\sigma}_i \in \mathscr{U}(\underline{m}_{\mathbf{ENC}}\ \underline{M}_{\mathbf{ENC}})$ (i.e. uniformly and randomly from the hypercube) and the corresponding reconstructed spectrum is calculated as $U\underline{\sigma}_i$. In Figure 5.12, panel (a), we show 3 of the randomly generated reflectances in **MOD**. Notice that they look like plausible spectra but there are – as expected – values less than 0 and greater than 1. Notice also that the reflectances are not completely smooth. This non smoothness 'local scale' appears to occur in nature, especially in the spectra of flowers and plants. Note, however, that, by construction, all the smooth spectra in Figure 5.4a (from the Macbeth ColorChecker) are in $box(\Omega_{\mathbf{MOD}})$.

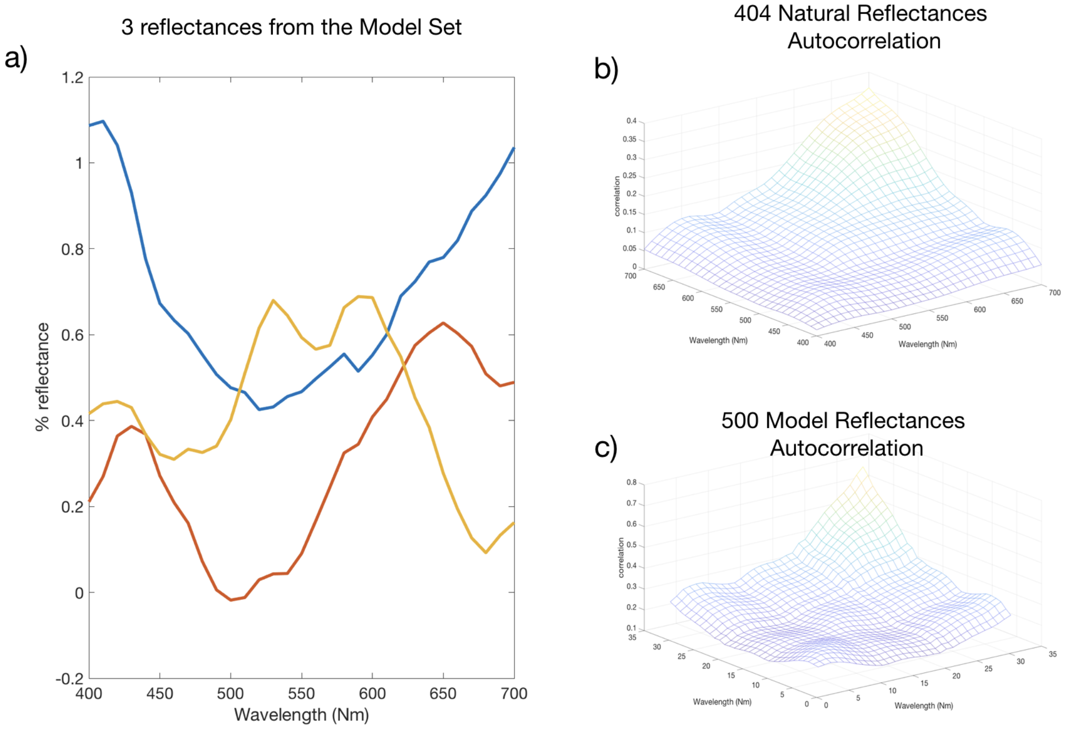


Fig. 5.12 In a), 3 reflectances from the Model reflectance dataset. Note there can be values larger than 1 and less than 0. Panels (b) and (c) show the autocorrelations for the **NAT** and **MOD** spectral datasets

We evaluated the extent to which the bounding hypercube representation admits implausible reflectance values i.e reflectances greater than 1 or less than 0. For out 5 data sets we found that more than 85% of our data are physically realizable.

Finally, in Figure 5.12 in panels (b) and (c) we respectively show the autocorrelations of **NAT** and **MOD**, see Figure 5.4 for the autocorrelations of the other reflectance sets.

	MUN	**OBJ**	**DUP**	**NAT**	**MOD**	μ
MUN	0	0.38	0.26	0.35	1.00	0.51
OBJ	0.57	0	0.76	0.44	2.10	0.96
DUP	0.23	0.44	0	0.30	0.80	0.44
NAT	0.38	0.32	0.37	0	1.25	0.58
MOD	0.52	0.68	0.45	0.57	0	0.56
μ	0.42	0.45	0.46	0.42	1.29	0.76

Table 5.1 $\frac{\%}{100}$ Reflectance Set Autocorrelation errors

5.5.2 Comparing the Autocorrelations of Reflectance Datasets

Let us begin by comparing directly all pairs of autocorrelations using a % error metric.

$$err_{A,B} = \frac{||auto(\Omega_A) - auto(\Omega_B)||}{||auto(\Omega_A)||} \tag{5.38}$$

Note, we represent the autocorrelations with respect to the Cosine Series Basis and the coordinates with respect to this basis. This said this error does not change if the data is mapped by an orthonornmal transform (so would be the same using the spectra themselves). The % autocorrelation error is not commutative. See Figure 5.10 and associated discussion.

The results for the 5 data sets are summarized in Table 1. The autocorrelations A and B from Equation 5.38 correspond to row and column respectively. The last column and row are respectively the average errors of the first 5 rows and columns. Bottom right is the overall average error.

Looking at the Table 5.1, errors range from 25% to over 200% with the overall average as 76%. However, remembering that **MOD**, by construction is designed to be a superset of the other 4 datasets, it is perhaps unsurprising that the comparisons that involve the autocorrelation of **MOD** are the ones that have the highest errors. Indeed, the average of the first 4 rows and columns is 40%. An interesting lesson we draw from the table is that the Dupont data – that comprises man-made dyed textiles – actually, has the lowest errors compared with the other data sets (see third row). That is, if we were to use only one of these datasets to represent the others (in terms of their autocorrelations), then we would choose the Dupont reflectances. This result is consistent with the colour correction experiments reported in Chapter 4,

In Table 5.2 we repeat the experiment for the autocorrelations of the enclosing hypercubes of the different datasets. Here, the datasets are labelled as **MUNb**, **OBJb**, **DUPb**, **NATb** and **MODb**, the **‘b’** stands for ’box’ (i.e. the figures are for the autocorrelations of the enclosing hypercube). The error is calculated as:

	MUNb	**OBJb**	**DUPb**	**NATb**	**MODb**	μ
MUNb	0	0.22	0.17	0.25	0.10	0.18
OBJb	0.27	0	0.12	0.44	0.42	0.28
DUPb	0.20	0.12	0	0.31	0.23	0.21
NATb	0.25	0.34	0.27	0	0.19	0.26
MODb	0.10	0.24	0.19	0.19	0	0.18
μ	0.20	0.23	0.19	0.29	0.20	0.28

Table 5.2 % Enclosing hypercube of Reflectance set autocorrelation errors

$$err_{A,B}^{Box} = \frac{||E(auto(box(\Omega_A))) - E(auto(box(\Omega_B)))||}{||E(auto(box(\Omega_A)))||} \tag{5.39}$$

With one exception (**NAT** compared with **OBJ**) the autocorrelations calculated over reflectance set's bounding hypercube are closer to each other (and usually much closer). Overall, the error is reduced from 76% to just 28% and the composite **MOD** set is now much closer to all other sets. Either the **MUN** and **MOD** would seem to be the best choice to represent all 5 datasets.

In the left of Figure 5.13 we show the autocorrelations of the sampled reflectance sets and on the right for the corresponding enclosing hypercubes. According to our measure of similarity the autocorrelations on the right are more similar than those on the left, on average. They do look more similar: they appear flatter overall. Notice also that the characteristic contours of the Munsell data (Figure 5.4) – are now replicated in the enclosing hypercube autocorrelations for the Object and Dupont datasets.

5.5.3 Encoding Reflectance Datasets

We might wonder, of course, what the meaning of a 50 to 60% reduction in autocorrelation error means? One way to answer this question is to think of the autocorrelation as a covariance matrix that captures the statistics of an underlying distribution. Indeed, suppose we used an autocorrelation to define a multivariate Gaussian (or Normal) distribution. By doing so we will admit many reflectances that lie outside even the bounding hypercube that we have laboured so long to develop. However, the reflectances that are drawn from this multivariate Gaussian will – by construction – have a covariance matrix equal to the given autocorrelation. Or, put another way reflectances distributed according to these implied Gaussian distributions would result in the same least-squares solution to colour correction.

Even although multivariate Gaussians are continuous functions, there exist analytic formulae that measure the number of bits on average – using an optimal encoding scheme

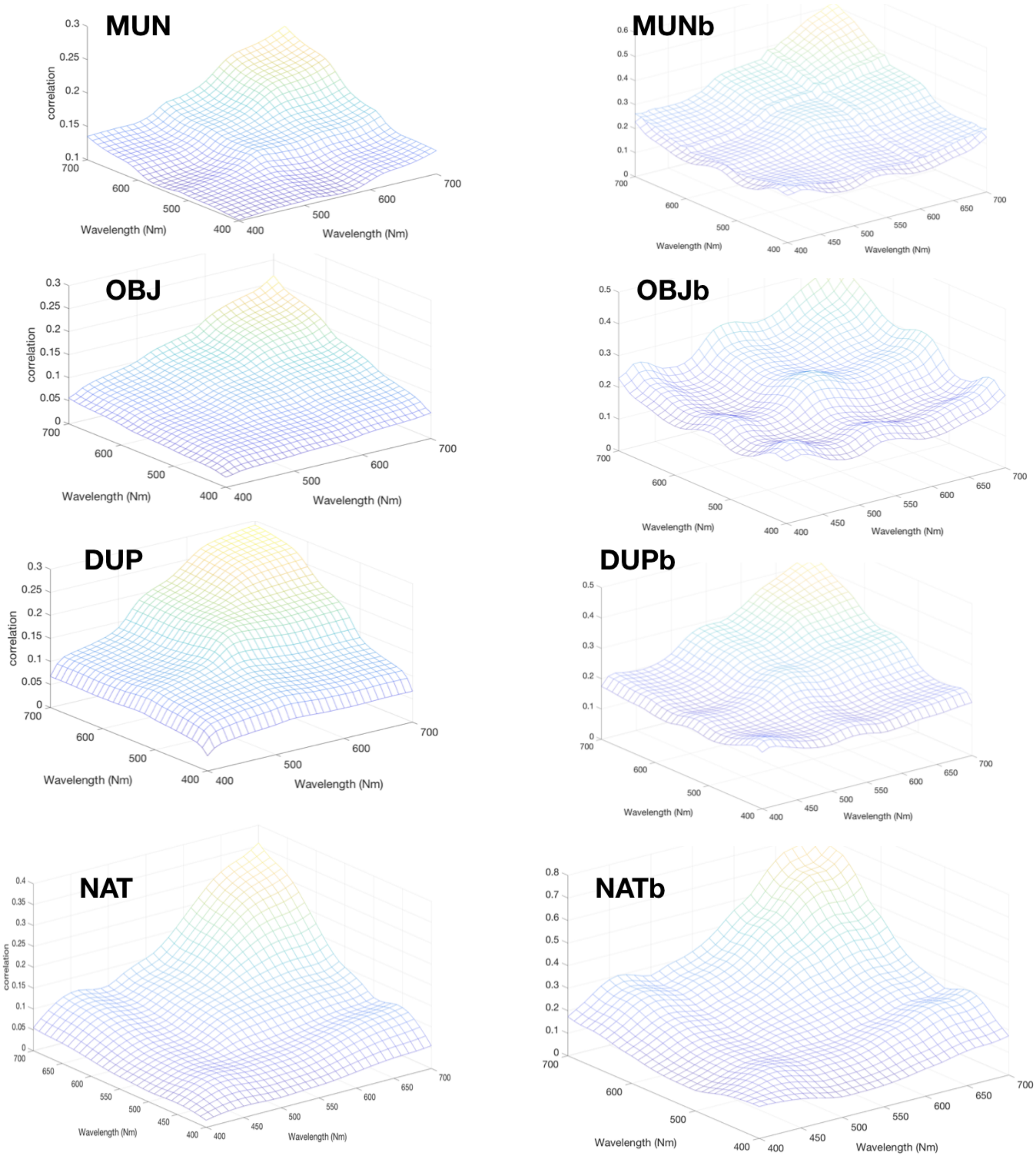


Fig. 5.13 On the left the 4 autocorrelation matrices and on the right the corresponding autocorrelation of the bounding box hypercubes

– it would take to represent a sample (here, a reflectance) drawn from that distribution. Remembering that a reflectance spectrum is a 31-dimensional vector, the ‘coding’ problem is not easy. Indeed, if, for example, we encoded the reflectance at each wavelength with an 8-bit number and we admit all spectra in [0,1] to be equally likely then we would need to use 31*8=248 bits to represent a spectrum. Of course, the actual bit count will be much less than this because reflectances are smooth (proximal wavelengths have similar values).

The average number of bits given an optimal encoding is called the entropy of a distribution. The link to ‘bit counting’ in the continuous domain is not direct and the numbers computed are only *weakly* related to entropies calculated for discrete measured data.

As commented earlier, for the purposes of colour correction the magnitude of the autocorrelations are not important. We can scale the autocorrelation matrix arbitrarily safe in the knowledge the scaling will cancel when we calculate the least-squares correction matrix. However, the scale also matters when we consider the number of bits it takes to code a Gaussian distribution. Henceforth the autocorrelation of the hypercube approximation is scaled to have the same mean value as the underlying reflectance distribution.

The issue here is one of ‘units’. As a tangential example suppose we wished to compare the distributions of the heights of children versus the heights of adults. Of course we cannot (adults are much taller). However, if we normalised the distributions – for example, computing the distribution z-scores[143] – we could compare the shape of the distributions. We can answer questions like: “are the proportions of children taller than mean height the same as for adults?” With the respect to the problem at hand, by matching the magnitude of covariance we are, informally, mapping different distributions to the same range for comparison.

In Table 5.3, we calculate the differential entropy in bits for distributions using our computed autocorrelations as covariances for Gaussian distributions (and assuming a 0 mean). We use the formulae for differential entropy [144] directly with one caveat. Here, we represent reflectances by the first 10 basis dimensions (which capture pretty much all the variance in all the datasets). Denoting the 1st 10 columns of our cosine series basis U by U^{10} and given a 31×31 auto-correlation (or an estimated autocorrelation) $AA^{\intercal}$ the 10×10 *equivalent* is equal to $[U^{10}]^{\intercal}AA^{\intercal}[U^{10}]$. We carry out this dimensionality reduction because the differential entropy formula does not work well if the covariance matrix is rank deficient. We found 10 basis functions sufficient to represent all the measured reflectance spectra

The numbers in Table 5.3 seem reasonable. For the raw data covariances we have an average entropy (number of bits to code a spectrum) of 33-bits. When we admit all reflectances in the enclosing hypercube then, intuitively, this should take more bits on

	MUN	**OBJ**	**DUP**	**NAT**	**MOD**	μ
entropy	26.5	29.2	29.7	33.1	44.6	32.6
box entropy	33.5	35.6	36.7	36.8	44.8	37.4

Table 5.3 The differential entropy (in bits) of the reflectance sets and their hypercube enclosures

	MUN	**OBJ**	**DUP**	**NAT**	**MOD**	μ
MUN	0	28.0	28.6	82.3	242.0	95.2
OBJ	9.2	0	9.4	7.3	73.7	24.9
DUP	35.5	14.5	0	32.4	210.4	72.5
NAT	11.5	3.4	6.9	0	41.4	15.8
MOD	12.6	9.7	10.3	6.8	0	8.8
μ	16.4	13.9	13.8	32.2	141.3	54.5

Table 5.4 The Kullback-Leibler divergences (all pairs reflectance sets).

average to encode. This is the case. To code the enclosing hypercube, on average takes 4.5 more bits than a direct coding of the underlying distribution.

Notice also that the **MOD** reflectance set has the highest entropy. This is expected as the **MOD** reflectances are drawn from the enclosing hypercube of all the reflectance data. And, of course because these reflectances are drawn uniformly and randomly.

The Kullback-Leibler divergence is one way of comparing distributions and complements the idea of entropy. Indeed, given two distributions A and B (and their optimal encodings), KL divergence measures the extra number of bits, on average, to code data from A using the coding from B (compared with using A's optimal encoding). Analogous to our % autocorrelation error the KL divergence measure is not commutative.

The KL-divergences for all pairs of our reflectance sets are summarized in Table 5.4. To orient the reader each row indicates the 'reference' distribution. As an example in position row=5, column=2, we have a KLD of 3.40 bits. This is the coding error of using the best codebook for the **MOD** data to encode the **OBJ** reflectance. As before, all computations assume we use only the first 10 terms in the cosine series basis to represent our reflectance data.

If we ignore the **MOD** data set, the coding error is in the range 3 to 80 bits . Of course the **MOD** – by construction – has reflectances quite unlike those in the other sets. So, the high coding errors are not surprising. Candidly, the numbers in Table 5.4 are a bit out of kilter to the magnitude of entropies computed in Table 5.3. Even with our 10-dimensional description our data is still rank deficient. The Munsells i particular are very smooth and so

	MUNb	**OBJb**	**DUPb**	**NATb**	**MODb**	μ
MUNb	0	4.17	1.89	3.41	22.96	8.10
OBJb	2.47	0	2.21	3.41	12.10	5.04
DUPb	1.07	1.56	0	1.41	9.25	3.32
NATb	1.98	2.31	1.52	0	8.81	3.66
MODb	5.99	4.12	3.52	3.41	0	4.28
μ	2.88	3.06	2.29	2.91	13.26	6.10

Table 5.5 The Kullback-Leibler divergences (all pairs reflectance sets defined by their bounding hypercubes)

the corresponding autocorrelation is somewhat rank deficient (concomitantly, the Munsell codebook isn't suitable to code the other distributions. Yet, if we go below 10 dimensions then some reflectances have high fitting errors. Table 5.4., while far from preesnting a clear narrative, represents a compromise between fairly representing all the datasets and mitigating against the rank-deficiency problem.

Having made these caveats, the numbers for the Kullback-Leibler divergences broadly make sense. We have an asymmetry between the coding error using **MOD** as a reference (e.g. as the code book that is not optimal that we are using) and when **MOD** encoded with another codebook. The row **MOD** teaches that this distribution can be used to encode other distributions with low error. This is as we expect since by construction **MOD** is a superset of the other data. Conversely, **MOD** attracts high errors when it is coded with the best codebook for other reflectance sets (which are small subsets of **MOD**), see column 5.

Let us repeat the experiment for the enclosing hypercubes of the reflectance sets. Here we look at the Kullback-Leibler divergence for all pairs of autocorrelations – deployed here as covariance of multivariate Gaussians – for the enclosing hypercubes of our data. As before **b** denotes 'bounding hypercube'.

Results are reported in Table 5.5. The KLD numbers are much smaller. This teaches both that in terms of coding the distributions are more similar and that they are *more* 10 dimensional (the KLD calculation is more stable).

5.5.4 Colour Correction Experiments

For a Nikon D300s camera we synthesised RGBs under D65. Similarly, we synthesized XYZs for the XYZ colour matching functions. We then computed the least-squares fit taking the RGBs to XYZs. Given the corresponding predicted and actual XYZs we calculated the average 'ΔE^*_{ab} ' error [33] for our 4 data sets (we ignore **MOD** because it has reflectances

	MUN	**OBJ**	**DUP**	**NAT**	μ
MUN	0.78	1.18	2.30	2.30	1.64
OBJ	1.45	1.11	2.47	1.79	1.70
DUP	1.28	0.94	1.71	1.44	1.34
NAT	2.00	1.25	2.23	1.28	1.69
μ	1.34	1.13	2.25	1.74	1.59

Table 5.6 Cross validated colour correction, mean ΔE^*_{ab}

	MUN	**OBJ**	**DUP**	**NAT**	μ
MUN	2.66	3.53	6.30	5.11	4.40
OBJ	3.13	3.38	9.30	4.11	4.98
DUP	2.66	2.46	4.87	3.64	3.41
NAT	3.27	3.04	5.75	3.35	3.85
μ	2.91	3.19	7.05	4.19	4.16

Table 5.7 Cross validated colour correction, 95 percentile ΔE^*_{ab}

that are not realizable. Mean and 95 percentile errors are shown in Tables 5.6 and 5.7. Each row ‘means’ we calculate the best 3×3 matrix using the autocorrelation for that surfaces reflectance set. We then test with all the reflectance sets in turn (each testing dataset represents a column).

The averages over columns are shown in the rightmost column. These averages encode how well a reflectance set performs when it is used to determine the colour correction transform. The average over rows speaks to the difficulty of correcting a given reflectance test set.

Clearly, when we train and test with the same reflectance set we get the best results – with the exception of **OBJ** to **OBJ**, due to the minimisation in **XYZ** and testing in **LAB** (the diagonal term has the lowest error per column). Training with the **DUP** set gives the lowest error overall, on average 1.34 ΔE^*_{ab} . Simultaneously the Dupont reflectances are the hardest to colour correct (an average ΔE^*_{ab} , 2.25). Note, that the mean error – of the whole table, shown bottom right is 1.59 – in general is fairly low (since 1 ΔE^*_{ab} is not perceptually significant).

In Table 5.7, we repeat the same experiment but tabulate the 95% error. As expected the 95% errors are significantly larger. Indeed, they are sufficiently large that the colour error of some patches could be noticeable in images. Note by a whisker using the **DUP** dataset to compute the colour correction matrix leads to the lowest average 95 percentile error for the

	MUN	**OBJ**	**DUP**	**NAT**	μ
MUNb	0.87	1.18	2.39	2.24	1.67
OBJb	0.95	1.07	2.37	2.02	1.60
DUPb	0.89	1.21	2.39	2.39	1.72
NATb	0.93	1.09	2.51	2.13	1.67
MODb	0.90	1.10	2.39	2.16	1.64
μ	0.91	1.13	2.41	2.18	1.66

Table 5.8 Cross validated colour correction, using enclosing hypercube to train, Mean ΔE^*_{ab}

	MUN	**OBJ**	**DUP**	**NAT**	μ
MUNb	2.86	3.86	5.99	5.03	4.43
OBJb	2.66	3.46	6.78	4.69	4.39
DUPb	2.79	3.81	6.47	5.49	4.64
NATb	2.54	3.76	8.80	5.14	5.06
MODb.	2.61	3.68	7.38	5.10	4.69
μ	2.62	3.71	7.08	5.09	4.64

Table 5.9 Cross validated colour correction, using enclosing hypercube to train, 95 percentile ΔE^*_{ab}

datasets (see third row). But, again, the Dupont reflectances themselves incur the highest percentile error, on average.

We now repeat this experiment where we train on the autocorrelation of all reflectances in the enclosing hypercube. We then test on the reflectance sets themselves. Here we do include the **MOD** dataset for training. Returning to Figure 5.9, the bounding hypercube of a reflectance set is many multiples in volume larger than the reflectances set itself (as defined by the convex closure). So, we are training on many more reflectances than the samples themselves. This said, we do not expect better colour correction results (certainly we must do worse than when we train and test on the same data). But if we obtain competitive performance then this means we can take a much more agnostic stance (about which reflectance appear in the world and still get good colour correction).

The mean colour correction results – when we train on the autocorrelation of the reflectance autocorrelation – are encouraging, see Table 5.8. We obtain similar results (indeed several are lower). On average (for corresponding entries) the error – already very low – remains low (about 15% higher) than fitting with the actual reflectances. Eight of the entries (there are 16 in common) are actually smaller).

The 95% results are shown in Table 5.9. Again results are comparable to training on the sample reflectance datasets (in this case we are better in 5 out of 16 cases). The penalty of training using the enclosing hypercube is less than 1 ΔE^*_{ab} (visually not significant).

We repeated this experiment for all 28 cameras in [88] and all illuminants in the 102 light set [97]. The trend of the data is the same as shown above.

5.6 Conclusion

In summary, in this chapter we have shown 3 results. First that the autocorrelations of two reflectance distributions are more similar if they are calculated over the enclosing hypercube (rather than just using the samples themselves). Second, similarity is also established by considering autocorrelations as covariances and looking at similarity as coding error (i.e. using the Kullback-Leibler divergence). Finally, we carried out a colour correction experiment. Solving for a colour correction using the enclosing hypercube of reflectance sets or the reflectance samples themselves leads to similar performance when evaluated on real sampled data.

Chapter 6

Maximum Ignorance Polynomial Colour Correction

6.1 Introduction

This chapter is an extended version of a published conference paper [145].

In the previous chapter (Chapter 5), we look at the question of reflectance dataset similarity in great depth. We argued that a reflectance dataset is a subsample of a much larger class of surfaces, and the convex closure of a reflectance set is possible. We then discussed how the convex closure of a set can be represented, and how this can be approximated by its enclosing hypercube. We finally show how we can integrate over our hypercube representation to calculate linear colour correction transforms. In this chapter, we expanded upon the idea of integration of the reflectance space further with an eye to using a more general colour correction transform. We consider allowing all possible reflectances - the Maximum Ignorance (MI) approach. The MI approach to colour correction assume that any spectrum, including the spectrum with negative values is equally likely to occur. Relative to this mathematical assumption it has been shown that the best colour correction matrix is the mapping which best takes the device specific spectral sensitivity functions onto the CIE XYZ colour matching functions [111]. This maximum ignorance transform is interesting as it also relates to the 'Luther' conditions which specify when perfect colour correction is possible. In particular, Horn [10] (and more recently Vora and Trussell [111]) has shown that perfect colour correction for any colour stimulus is possible if and only the Luther conditions are met, i.e. when the device sensitivities are a linear transform from the colour matching functions.

Arguably, however, for cameras that do not meet the Luther conditions, the MI assumption is practically not useful because it does not make physical sense. Indeed, spectra with negative power do not exist in nature, so making the assumption that they do exist has negative impact on the performance of colour correction. Finlayson and Drew addressed this problem by introducing the concept of maximum ignorance with positivity (MIP) [112]. Under this assumption, all colour signals are assumed to be strictly positive and occur with equal likelihood. The colour signal is drawn uniformly and randomly from the interval (where is an upper bound on the power at any wavelength). Finlayson and Drew presented an algorithm for MIP colour correction [112] which they showed depended only on the spectral sensitivities of the sensors (RGB and XYZ) and the autocorrelation of the spectra (which they computed in closed form for the MIP assumption). Experiments demonstrated that the MIP approach delivered much better colour correction than the MI method. Indeed, the results were found to be comparable to the physical target based approaches.

In this chapter, we revisit and extend the concept of MIP, we make two contribution. Our first contribution is reformulating the computation of the colour correction matrix in terms of sensor response rather than spectral correlations. The advantage of doing so is that it allows us to consider non-linear correction schemes. In our second contribution, we show how we can derive the polynomial regression matrix given maximum ignorance with positivity assumptions (we call this MIPP). The practical importance of this work is also considered.

6.2 Theory

6.2.1 Maximum Ignorance with Positivity Colour Correction

Suppose a colour signal $C(\lambda)$ is represented by discrete samples in the visible spectrum – between 400 nm to 700 nm at 10 nm intervals, with 31 discrete samples in total). Let $\mathbf{C}$ denote the $31 \times n$ matrix containing a set of n calibration colour signal spectra (with one colour signal per column). Let $\mathbf{X}$ and $\mathbf{R}$ denote the 31×3 matrices containing the CIE XYZ standard observer colour matching function and device spectral sensitivities. The camera and human observer response to the entire calibration set are captured by $3 \times n$ matrices $\mathbf{P}$ and $\mathbf{Q}$ [146]:

$$\mathbf{P} = \mathbf{R}^{\mathsf{T}}\mathbf{C} \tag{6.1}$$

$$\mathbf{Q} = \mathbf{X}^{\mathsf{T}}\mathbf{C} \tag{6.2}$$

In colour correction we wish to map mathematically $\mathbf{P}$ to $\mathbf{Q}$. The least-squares solution to colour correction finds 3×3 matrix $\mathbf{M}$ which minimises:

$$||\mathbf{MP} - \mathbf{Q}||_F \tag{6.3}$$

$||.||_F$ above denotes the Frobenius norm (the square root of the sum of squared differences between $\mathbf{MP}$ and $\mathbf{Q}$). The Matrix $\mathbf{M}$ which minimises Equation 6.3 is found in closed-form using the Moore–Penrose pseudoinverse:

$$\mathbf{M} = \mathbf{QP}^{\mathsf{T}}(\mathbf{PP}^{\mathsf{T}})^{-1} \tag{6.4}$$

By substituting Equation 6.1 and Equation 6.2 into Equation 6.4, we obtain:

$$\mathbf{M} = \mathbf{X}^{\mathsf{T}}\mathbf{CC}^{\mathsf{T}}(\mathbf{R}^{\mathsf{T}}\mathbf{CC}^{\mathsf{T}}\mathbf{R})^{-1} \tag{6.5}$$

We can see from Equation 6.5 that $\mathbf{M}$ depends only on the 31×3 device spectral sensitivities $\mathbf{R}$, the 31×3 CIE standard observer colour matching function $\mathbf{X}$ and the 31×31 spectral autocorrelation matrix $\mathbf{CC}^{\mathsf{T}}$.

In MIP, the spectral autocorrelation matrix is artificially constructed to enforce positive autocorrelation. Suppose we wish to represent all possible colour signal spectra in the interval $[0, P]$, where P denotes the maximum spectral power per wavelength. Because the spectral autocorrelation matrix and its inverse both appear in Equation 6.5, the magnitude of spectral autocorrelation matrix is not important. Hence without losing generality, we can assume spectra lie in the interval $[0, 1]$. Under these conditions, according to [112] $\mathbf{CC}^{\mathsf{T}}$ equals to:

$$\left[\mathbf{CC}^{\mathsf{T}}\right]_{ij} = \begin{cases} \frac{1}{3}\ (i = j) \\ \\ \frac{1}{4}\ (i \neq j) \end{cases}, \tag{6.6}$$

Our new formulation is based on the expectation of the sensor response correlation. In order to tackle this problem, we first look at how to compute the expectation of the response of a sensor with a single colour channel. Let $\mathbf{r}$ denote the 31-vector containing sensor response curve from a single colour channel. Let $\mathbf{c}$ denote the 31-vector colour signal. The sensor response p is computed as a dot-product:

$$\begin{aligned} p &= \mathbf{r}^{\mathsf{T}}\mathbf{c} \\ &= \mathbf{r}\cdot\mathbf{c} \\ &= r_1c_1 + r_2c_2 + \cdots + r_{31}c_{31} \end{aligned} \tag{6.7}$$

Let us assume that each colour signal sample is an independent and identically distributed random variable with values between 0 and 1. The *expected* value of p, $\mathbb{E}(p)$ is written as:

$$\begin{aligned} &\mathbb{E}(p) \\ &= \int_0^1 \mathbf{r}\cdot\mathbf{c} \quad d\mathbf{c} \\ &= \int_0^1 \cdots \int_0^1 r_1c_1 + r_2c_2 + \cdots + r_{31}c_{31} \quad \mathrm{d}c_1 \cdots \mathrm{d}c_{31} \end{aligned} \tag{6.8}$$

We now need to apply the idea from Equation 6.8 to Equation 6.4. Now, let us explicitly write the least-squares matrix calculation in terms of the correlations of sensor responses:

$$\begin{aligned} \mathbf{M} &= \mathbf{Q}\mathbf{P}^{\mathsf{T}}(\mathbf{P}\mathbf{P}^{\mathsf{T}})^{-1} \\ &= \begin{bmatrix} \mathbf{Q}_1\mathbf{P}_1^{\mathsf{T}} & \mathbf{Q}_1\mathbf{P}_2^{\mathsf{T}} & \mathbf{Q}_1\mathbf{P}_3^{\mathsf{T}} \\ \mathbf{Q}_2\mathbf{P}_1^{\mathsf{T}} & \mathbf{Q}_2\mathbf{P}_2^{\mathsf{T}} & \mathbf{Q}_2\mathbf{P}_3^{\mathsf{T}} \\ \mathbf{Q}_3\mathbf{P}_1^{\mathsf{T}} & \mathbf{Q}_3\mathbf{P}_2^{\mathsf{T}} & \mathbf{Q}_3\mathbf{P}_3^{\mathsf{T}} \end{bmatrix} \begin{bmatrix} \mathbf{P}_1\mathbf{P}_1^{\mathsf{T}} & \mathbf{P}_1\mathbf{P}_2^{\mathsf{T}} & \mathbf{P}_1\mathbf{P}_3^{\mathsf{T}} \\ \mathbf{P}_2\mathbf{P}_1^{\mathsf{T}} & \mathbf{P}_2\mathbf{P}_2^{\mathsf{T}} & \mathbf{P}_2\mathbf{P}_3^{\mathsf{T}} \\ \mathbf{P}_3\mathbf{P}_1^{\mathsf{T}} & \mathbf{P}_3\mathbf{P}_2^{\mathsf{T}} & \mathbf{P}_3\mathbf{P}_3^{\mathsf{T}} \end{bmatrix}^{-1} \end{aligned} \tag{6.9}$$

In terms of Equation 6.9, we would like to compute the expected values of $\mathbf{Q}\mathbf{P}^{\mathsf{T}}$ and $\mathbf{P}\mathbf{P}^{\mathsf{T}}$. The terms in these two matrices can be computed if for arbitrary matrices $\mathbf{X}$ and $\mathbf{Y}$ we can compute $(\mathbf{X}\mathbf{Y})_{ij}$. Denoting the i^{th} row of $\mathbf{X}$ as the vector $\boldsymbol{\alpha}$ and the j^{th} column of $\mathbf{Y}$ as $\boldsymbol{\beta}$, $\mathbb{E}(\mathbf{X}\mathbf{Y})_{ij}$ can be computed by solving the following equation:

$$\begin{aligned} &\mathbb{E}(\mathbf{X}\mathbf{Y})_{ij} \\ &= \mathbb{E}((\boldsymbol{\alpha}\cdot\mathbf{c})(\boldsymbol{\beta}\cdot\mathbf{c})) \\ &= \int_0^1 \cdots \int_0^1 \left(\sum_{i=1}^{31} c_i\alpha_i\right)\left(\sum_{i=1}^{31} c_i\beta_i\right) \mathrm{d}c_1 \cdots \mathrm{d}c_{31} \end{aligned} \tag{6.10}$$

To demonstrate our formula, for illustrative purposes, let us assume that our sensor takes samples at two discrete wavelengths:

$$\begin{aligned}
&\mathbb{E}((\boldsymbol{\alpha}\cdot\mathbf{c})(\boldsymbol{\beta}\cdot\mathbf{c}) \\
&= \int_0^1\int_0^1 (c_1\alpha_1+c_2\alpha_2)(c_1\beta_1+c_2\beta_2)\ \mathrm{d}_1\ \mathrm{d}_2 \\
&= \int_0^1\int_0^1 c_1^2\alpha_1\beta_1+(\alpha_1\beta_2+\alpha_2\beta_1)c_1c_2+\alpha_2\beta_2c_2^2\ \mathrm{d}_1\ \mathrm{d}_2 \\
&= \frac{\alpha_1\beta_1}{3}+\frac{\alpha_2\beta_1}{4}+\frac{\alpha_2\beta_2}{3}
\end{aligned} \tag{6.11}$$

Equation (6.11) can be extended to include 31 wavelengths, the derivations are not provided here.

6.2.2 Polynomial Maximum Ignorance with Positivity Colour Correction (MIPP)

In this section, we extend Maximum Ignorance colour correction, and so it can be applied second order polynomial colour correction. We achieve this by adding ‘squared’ and ‘cross’ terms to each RGB camera measurement: each input RGB is mapped to a 9-vector: $(r,g,b,r^2,g^2,b^2,rg,rb,gb)^\intercal$. We can then again use Moore-Penrose inverse to solve for the colour transformation, although here $\mathbf{QP}^\intercal$ and $\mathbf{PP}^\intercal$ have the dimensions of 3×9 and 9×9 respectively.

The expectation for the cross-product terms between the linear terms and polynomial terms in the 3×9 matrix $\mathbf{QP}^\intercal$ are calculated using the following equation:

$$\begin{aligned}
&\mathbb{E}((\boldsymbol{\alpha}\cdot\mathbf{c})(\boldsymbol{\beta}\cdot\mathbf{c})(\boldsymbol{\gamma}\cdot\mathbf{c})) \\
&= \int_0^1\cdots\int_0^1 \left(\sum_{i=1}^{31} c_i\alpha_i\right)\left(\sum_{i=1}^{31} c_i\beta_i\right)\left(\sum_{i=1}^{31} c_i\gamma_i\right)\ \mathrm{d}c_1\cdots\mathrm{d}_{31}
\end{aligned} \tag{6.12}$$

As an example, to compute $\mathbb{E}((\mathbf{r}\cdot\mathbf{c})(\mathbf{r}\cdot\mathbf{c})(\mathbf{x}\cdot\mathbf{c}))$, the $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ in Equation 6.12 are redefined as the sensitivities of imaging sensor’s red channel, and γ is redefined as the X channel of the CIE XYZ colour matching function.

The expectation for the auto-product terms in the 9×9 matrix $\mathbf{PP}^\mathsf{T}$ can be computed using the following equation:

$$\begin{aligned}&\mathbb{E}((\boldsymbol{\alpha}\cdot\mathbf{c})(\boldsymbol{\beta}\cdot\mathbf{c})(\boldsymbol{\gamma}\cdot\mathbf{c})(\boldsymbol{\delta}\cdot\mathbf{c}))\\ &=\int_0^1\cdots\int_0^1\left(\sum_{i=1}^{31}c_i\alpha_i\right)\left(\sum_{i=1}^{31}c_i\beta_i\right)\left(\sum_{i=1}^{31}c_i\gamma_i\right)\left(\sum_{i=1}^{31}c_i\delta_i\right)\mathrm{d}c_1\cdots\mathrm{d}_{31}\end{aligned} \tag{6.13}$$

As an example, to compute $\mathbb{E}((\mathbf{c}\cdot\mathbf{r})(\mathbf{c}\cdot\mathbf{r})(\mathbf{c}\cdot\mathbf{g})(\mathbf{c}\cdot\mathbf{b}))$, we substitute $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ in Equation 6.13 with the spectral sensitivity of the red channel of the imaging sensor. We substitute $\boldsymbol{\gamma}$ and $\boldsymbol{\delta}$ with the spectral sensitivities of the green and blue channels of the imaging sensor respectively.

6.3 Experiment

We performed colour correction experiments using a Nikon D5100 camera. The spectral sensitivity measurement data for the camera can be found in [89]. Relative to this camera, we computed the colour correction matrices for the following methods: Maximum Ignorance, Maximum Ignorance with Positivity and Polynomial Maximum Ignorance with Positivity. This includes two 3×3 matrices, and a 3×9 matrix.

6.3.1 Simulation experiment using synthetic colour signal

We generated colour signals using all pairs of 102 illuminant spectra and 1995 reflectances [97]. In total, there were over 200,000 spectra in our colour signal dataset. We calculated their RGBs and corresponding XYZs by numerical integration. We applied each of the 3 Maximum Ignorance colour correction matrices to the camera RGBs in order to obtain the estimated XYZs. We then compared the estimated XYZs with true XYZs using CIELAB ΔE^*_{ab} . The results are shown in Table 6.1.

Table 6.1 The CIELAB ΔE^*_{ab} for colour correction experiment using synthetic data

Method	**Mean**	**Median**	**95%**
Maximum Ignorance	5.25	3.54	13.02
Maximum Ignorance with Positivity	**3.16**	**2.14**	**8.64**
Polynomial Maximum Ignorance with Positivity	4.52	3.48	11.31

6.3.2 Experiment using real camera data

We also performed colour correction experiment using real camera data. We used a 24-patch X-Rite ColorChecker Classic as our training target. Under cloudy daylight, the radiance from each patch of the training target was measured using a Photo Research PR-670. By numerically integrating the radiance with the CIE XYZ colour matching function, we obtained the true XYZs of the target patches. Great care was taken with the measurement geometry to ensure that each patch was measured in the same way. We used the raw images output from Nikon D5100 as the RGB values for the colour patches. The effect of shading variation across the ColorChecker was normalised by dividing the RGB values by brightness values from an image of a flat matt-white reflector.

Table 6.2 The CIELAB ΔE^*_{ab} for colour correction experiment involving real world data

Method	**Mean**	**Median**	**95%**
Maximum Ignorance	5.32	4.46	12.22
Maximum Ignorance with Positivity	**3.99**	**3.66**	**8.25**
Polynomial Maximum Ignorance with Positivity	6.25	5.02	12.5

6.4 Conclusion

In *linear* colour correction, Maximum Ignorance with Positivity assumes that all possible positive spectra are equally likely. Relative to this assumption, colour correction depends only on the autocorrelation of the spectra and the device spectral sensitivities. However, to apply the MIP assumption to *non-linear* colour correction is much more complex. One contributions of this chapter is to show how we can solve the problem of polynomial regression under the MIP assumption, by presenting the mathematics behind the modelling of the expected sensor response.

Experiments, however, demonstrated that the polynomial maximum ignorance assumption works less well than the antecedent methods. Speculatively, this underperformance can be explained by the higher order terms for ‘unlikely’ sharp spectra dominating the regression. Further investigation can be conducted by computing the autocorrelation matrix in subsection 6.2.1 without the positivity constraint and compare it to autocorrelation matrix with the constraints. The comparison metrics can be Kullback Leibler divergence and encoding entropy, similar to the approaches in subsection 5.5.3

The observation that we might examine performance for all spectra in terms of expected sensor responses is one that can be extended to other colour correction scenarios which we are investigating. Overall, the derivation performed in this chapter might be of interest for those who are interested in taking the integration approach for computing colour correction further.

Chapter 7

Conclusion

In this thesis, we explored the topic of colour correction.

In Chapter 2, we started off by giving an overview of the Human Visual System and digital cameras, with an emphasis on how colour responses are formed. Many of the mathematical tools that we employed throughout the thesis – and in particular linear algebra – are introduced.

In Chapter 3, we provided a review on various colour correction algorithms. We discussed the ideas behind the algorithms and their advantages and disadvantages. A special emphasis was placed on reviewing ‘exposure invariance’. This is the idea that if we simply change the amount of light present in a scene some algorithms stop working. For general colour correction we propose it is important to maintain exposure invariance (since we do not wish an algorithm to stop working because the brightness has changed). Finally we evaluated the empirical performances of the current state of the art in colour correction.

As a companion to Chapter 3 we developed a Matlab toolbox where many of the leading colour correction algorithms have been implemented.

In Chapter 4, we provided a comparison of reflectance datasets. We attempted to address the question of "which dataset should be used for training a colour correction algorithm?". The fact that we ultimately failed to answer this question was both the motivation and springboard for Chapter 5.

Chapter 4 presented results for commonly used measures of similarity in a cross-validation sense, i.e. how well can one dataset be used to train the colour correction transform and how well this transform works when applied to a different dataset. We also presented new metrics. In work that is related to Chapter 5 we considered representing the colours produced by a reflectance dataset by its convex closure. We also presented a novel application of the Vora Value metric. The Vora Value measure is used to evaluate the similarity between two sets of camera. Via a reflectance basis decomposition, we use it to compare two reflectance datasets. In complete alignment with previous unrelated studies our investigation of the Vora Value measure led us to the conclusion that reflectances can be considered as living in 8 or 9 dimensional subspace.

Finally, in a surprising empirical result, we showed that the data set that is most difficult to colour correct is simultaneously the one that leads to the best cross-validated colour correction performance.

Chapter 5 presents a major contribution of this thesis. We revisited the convex closure idea presented in Chapter 4. There, it was argued that because reflectances viewed in combination (e.g. when viewed from distance) merge in proportion to their relative proportions, all the reflectances in the convex closure of a reflectance set are possible.

However, computationally, a convex closure (even with a 9-dimensional basis representation) is an unwieldy structure. We therefore proposed to represent the convex closure by its hypercube enclosure (or bounding box). However, we demonstrated that it was important that the bounding box was calculated with respect to the correct coordinate system. Here the idea of a *common basis* is important. We represented all reflectances by their expansion (by their coordinates) with respect to the cosine series basis. The bounding box with respect to the common basis is a *closer* fit compared to the bounding box of the reflectances themselves (i.e. functions of wavelength).

The hypercube enclosure is a continuous representation of a reflectance set. Therefore algorithms are developed to calculate a colour correction transform given our hypercube representation. Experiments demonstrate that the hypercube enclosure of reflectance sets drives good colour correction and also works under a cross-validation evaluation. Significantly, in terms of impact on colour correction, the enclosing hypercubes of different reflectance sets are much more similar than the reflectance datasets themselves. To some extent we answer the question “what reflectance dataset should I use?” with the answer “it doesn’t matter so long as you use its convex enclosure”. One of the disadvantages of Chapter 5 is that to use the bounding hypercube idea for colour correction one has to use linear regression (it is the only method that seems feasible given a continuous set of reflectances).

In Chapter 6, we expanded upon the idea of integration of the reflectance space further with an eye to using a more general colour correction transform. We considered allowing all possible reflectances. This is the so-called MIP: Maximum ignorance with Positivity Assumption. Relative to this simple MIP assumption, we showed how colour correction algorithm more advanced than linear least squared could be trained and used. This is a theoretical contribution, however. The MIP assumptuon based polynomial colour correction does not perform better than linear regression.

7.1 Future work

The idea of reflectance space integration can be definitely explored further.

For training colour correction matrices, larger datasets are typically considered to be better, as they cover a wider range of colours. In Chapter 5, when we calculated the autocorrelation matrix by integration, the parameters we took into account were the limit of the bounding box, and the basis of the reflectance space. One avenue of research would be projecting the limits of the bounding box onto the basis, and using the resulting reflectances to training colour correction matrix. We effectively resample a large reflectance dataset using a small number of samples. It would be interesting to see the smaller reflectance dataset can achieve similar performance as the larger reflectance dataset. This research could potentially lead to a new compact colour checker.

Another limitation to our approach of calculating the reflectance dataset autocorrelation is that we only take the range of the sampled reflectance within the reflectance space into account. We do not take the distribution of sample dataset into account. It would be interesting to incorporate the distribution statistics of the dataset when calculating the autocorrelation matrix using integration. Statistics such as mean and variance affect the final shape of the autocorrelation matrix.

It is not very clear why polynomial maximum ignorance colour correction has poor real-world performance. Further investigation would be fruitful. It would be interesting to derive the root-polynomial variant of the maximum ignorance colour correction algorithm. Compared to polynomial colour correction, root-polynomial colour correction has the property of being exposure invariant. Root-polynomial maximum ignorance colour correction would therefore be useful in real-world conditions, assuming its performance is adequate.

References

[1] University of Texas, “The first photograph,” https://www.hrc.utexas.edu/exhibitions/permanent/firstphotograph/#top/heliography.html, (Accessed on 03/07/2019).

[2] R. V. Jones, “James clerk maxwell at aberdeen-1856-1860,” *Notes and Records of the Royal Society of London*, vol. 28, no. 1, pp. 57–81, 1973.

[3] J. Nakamura, *Image sensors and signal processing for digital still cameras*. CRC press, 2016.

[4] S. Temple, “Vintage mobiles | gsm history: History of gsm, mobile networks, vintage mobiles,” http://www.gsmhistory.com/vintage-mobiles/#sharp_sh04_2001, (Accessed on 03/08/2019).

[5] C. Jarvis, *The Best Camera Is the One That’s with You: IPhone Photography*, ser. Voices That Matter Series. New Riders, 2010. [Online]. Available: https://books.google.co.uk/books?id=TwppjgEACAAJ

[6] B. C. Regan, C. Julliot, B. Simmen, F. Vienot, P. Charles-Dominique, and J. D. Mollon, “Fruits, foliage and the evolution of primate colour vision,” *Philosophical Transactions of the Royal Society B: Biological Sciences*, vol. 356, no. 1407, pp. 229–283, 2001.

[7] M. O’sullivan, D. Byrne, H. Martens, L. Gidskehaug, H. Andersen, and M. Martens, “Evaluation of pork colour: prediction of visual sensory quality of meat from instrumental and computer vision methods of colour analysis,” *Meat Science*, vol. 65, no. 2, pp. 909–918, 2003.

[8] C. Costa, F. Antonucci, P. Menesatti, F. Pallottino, C. Boglione, and S. Cataudella, “An advanced colour calibration method for fish freshness assessment: A comparison between standard and passive refrigeration modalities,” *Food and Bioprocess Technology*, vol. 6, no. 8, pp. 2190–2195, 2013.

[9] P. Sumner and J. Mollon, “Chromaticity as a signal of ripeness in fruits taken by primates,” *Journal of Experimental Biology*, vol. 203, no. 13, pp. 1987–2000, 2000.

[10] B. K. Horn, “Exact reproduction of colored images,” *Computer Vision, Graphics, and Image Processing*, vol. 26, no. 2, pp. 135–167, 1984.

[11] W. Stiles and J. Burch, “Interim report to the commission internationale de l’eclairage, zurich, 1955, on the national physical laboratory’s investigation of colour-matching (1955),” *Journal of Modern Optics*, vol. 2, no. 4, pp. 168–181, 1955.

[12] C.-I. Chang, *Hyperspectral imaging: techniques for spectral detection and classification*. Springer Science & Business Media, 2003, vol. 1.

[13] G. Wald, "Human vision and the spectrum." *Science*, 1945.

[14] R. Ramanath, W. E. Snyder, Y. Yoo, and M. S. Drew, "Color image processing pipeline," *IEEE Signal Processing Magazine*, vol. 22, no. 1, pp. 34–43, 2005.

[15] M. Tkalcic, J. F. Tasic *et al.*, "Colour spaces: perceptual, historical and applicational background," in *Eurocon*, 2003.

[16] K. E. Spaulding, G. J. Woolfe, and E. J. Giorgianni, "'reference input/output medium metric rgb color encodings," *Proc. IS&T PICS*, pp. 155–163, 2000.

[17] S. Süsstrunk, R. Buckley, and S. Swen, "Standard rgb color spaces," in *Color and Imaging Conference*, vol. 1999, no. 1. Society for Imaging Science and Technology, 1999, pp. 127–134.

[18] W. Commons, "File:eyesection.svg — wikimedia commons, the free media repository," 2018, [Online; accessed 18-March-2019]. [Online]. Available: https://commons.wikimedia.org/w/index.php?title=File:Eyesection.svg&oldid=286009081

[19] A. Roorda, F. Romero-Borja, W. J. Donnelly III, H. Queener, T. J. Hebert, and M. C. Campbell, "Adaptive optics scanning laser ophthalmoscopy," *Optics express*, vol. 10, no. 9, pp. 405–412, 2002.

[20] J. Roufs, "Light as a true visual quantity: Principles of measurement," 1978.

[21] G. Wyszecki and W. S. Stiles, *Color science*. Wiley New York, 1982, vol. 8.

[22] A. Roorda and D. R. Williams, "The arrangement of the three cone classes in the living human eye," *Nature*, vol. 397, no. 6719, p. 520, 1999.

[23] A. Stockman, D. I. MacLeod, and N. E. Johnson, "Spectral sensitivities of the human cones," *JOSA A*, vol. 10, no. 12, pp. 2491–2521, 1993.

[24] T. Young, "Ii. the bakerian lecture. on the theory of light and colours," *Philosophical transactions of the Royal Society of London*, no. 92, pp. 12–48, 1802.

[25] J. C. Maxwell, "Xviii.—experiments on colour, as perceived by the eye, with remarks on colour-blindness," *Earth and Environmental Science Transactions of the Royal Society of Edinburgh*, vol. 21, no. 2, pp. 275–298, 1857.

[26] M. S. Longair, "Maxwell and the science of colour," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 366, no. 1871, pp. 1685–1696, 2008.

[27] W. D. Wright, "A re-determination of the trichromatic coefficients of the spectral colours," *Transactions of the Optical Society*, vol. 30, no. 4, p. 141, 1929.

[28] J. Guild, "The colorimetric properties of the spectrum," *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, vol. 230, pp. 149–187, 1932.

[29] A. Giannini and L. Mercatelli, *Colorimetric Quantities and Laws*. London: Springer London, 2015, pp. 23–37. [Online]. Available: https://doi.org/10.1007/978-1-4471-6633-7_2

[30] Y. Ohno, "Cie fundamentals for color measurements," in *NIP & Digital Fabrication Conference*, vol. 2000, no. 2. Society for Imaging Science and Technology, 2000, pp. 540–545.

[31] G. W. Meyer and D. P. Greenberg, "Perceptual color spaces for computer graphics," *ACM SIGGRAPH Computer Graphics*, vol. 14, no. 3, pp. 254–261, 1980.

[32] D. L. MacAdam, "Visual sensitivities to color differences in daylight," *JOSA*, vol. 32, no. 5, pp. 247–274, 1942.

[33] M. Mahy, L. Eycken, and A. Oosterlinck, "Evaluation of uniform color spaces developed after the adoption of cielab and cieluv," *Color Research & Application*, vol. 19, no. 2, pp. 105–121, 1994.

[34] G. Paschos, "Perceptually uniform color spaces for color texture analysis: an empirical evaluation," *IEEE transactions on Image Processing*, vol. 10, no. 6, pp. 932–937, 2001.

[35] J. Schanda, *Colorimetry: understanding the CIE system*. John Wiley & Sons, 2007.

[36] "CIE Publication 116: Industrial colour-difference evaluation," Commission internationale de l'éclairage, Vienna AUSTRIA, Tech. Rep., 1995.

[37] M. Melgosa, "Testing CIELAB-based color-difference formulas," *Color Research & Application*, vol. 25, no. 1, pp. 49–55, 2000.

[38] M. R. Luo, G. Cui, and B. Rigg, "The development of the CIE 2000 colour-difference formula: CIEDE2000," *Color Research & Application*, vol. 26, no. 5, pp. 340–350, 2001.

[39] G. C. Holst and T. S. Lomheim, *CMOS/CCD sensors and camera systems*. JCD Publishing USA, 2007, vol. 408.

[40] A. Kinba, M. Hamada, H. Ueda, K. Sugitani, and H. Ootsuka, "Auto focus detecting device comprising both phase-difference detecting and contrast detecting methods," Jan. 28 1997, US Patent 5,597,999.

[41] T. Kunkel and E. Reinhard, "A reassessment of the simultaneous dynamic range of the human visual system," in *Proceedings of the 7th Symposium on Applied Perception in Graphics and Visualization*. ACM, 2010, pp. 17–24.

[42] M. A. Robertson, S. Borman, and R. L. Stevenson, "Dynamic range improvement through multiple exposures," in *Proceedings 1999 International Conference on Image Processing (Cat. 99CH36348)*, vol. 3. IEEE, 1999, pp. 159–163.

[43] E. Reinhard, W. Heidrich, P. Debevec, S. Pattanaik, G. Ward, and K. Myszkowski, *High dynamic range imaging: acquisition, display, and image-based lighting*. Morgan Kaufmann, 2010.

[44] D. Litwiller, “Ccd vs. cmos,” *Photonics Spectra*, vol. 35, no. 1, pp. 154–158, 2001.

[45] “Camera feature search: Digital photography review,” https://www.dpreview.com/products/search/cameras, (Accessed on 04/03/2019).

[46] “File:jamtlands flyg ec120b colibri.jpg - wikipedia,” https://en.wikipedia.org/wiki/File:Jamtlands_Flyg_EC120B_Colibri.JPG, (Accessed on 04/03/2019).

[47] B. E. Bayer, “Color imaging array,” Jul. 20 1976, US Patent 3,971,065.

[48] X. Li, B. Gunturk, and L. Zhang, “Image demosaicing: A systematic survey,” in *Visual Communications and Image Processing 2008*, vol. 6822. International Society for Optics and Photonics, 2008, p. 68221J.

[49] B. K. Gunturk, J. Glotzbach, Y. Altunbasak, R. W. Schafer, and R. M. Mersereau, “Demosaicking: color filter array interpolation,” *IEEE Signal processing magazine*, vol. 22, no. 1, pp. 44–54, 2005.

[50] R. Kimmel, “Demosaicing: image reconstruction from color ccd samples,” *IEEE Transactions on image processing*, vol. 8, no. 9, pp. 1221–1228, 1999.

[51] K. Hirakawa and T. W. Parks, “Adaptive homogeneity-directed demosaicing algorithm,” *IEEE Transactions on Image Processing*, vol. 14, no. 3, pp. 360–369, 2005.

[52] X. Li, “Demosaicing by successive approximation,” *IEEE Transactions on Image Processing*, vol. 14, no. 3, pp. 370–379, 2005.

[53] E. P. Bennett, M. Uyttendaele, C. L. Zitnick, R. Szeliski, and S. B. Kang, “Video and image bayesian demosaicing with a two color image prior,” in *European Conference on Computer Vision*. Springer, 2006, pp. 508–521.

[54] O. Kapah and H. Z. Hel-Or, “Demosaicking using artificial neural networks,” in *Applications of Artificial Neural Networks in Image Processing V*, vol. 3962. International Society for Optics and Photonics, 2000, pp. 112–121.

[55] Y.-Q. Wang, “A multilayer neural network for image demosaicking,” in *2014 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2014, pp. 1852–1856.

[56] R. Ramanath, W. E. Snyder, G. L. Bilbro, and W. A. Sander, “Demosaicking methods for bayer color arrays,” *Journal of Electronic imaging*, vol. 11, no. 3, pp. 306–316, 2002.

[57] B. Dierickx and G. Meynants, “Missing pixel correction algorithm for image sensors,” in *Advanced Focal Plane Arrays and Electronic Cameras II*, vol. 3410. International Society for Optics and Photonics, 1998, pp. 200–204.

[58] S. Kavadias, B. Dierickx, D. Scheffer, A. Alaerts, D. Uwaerts, and J. Bogaerts, “A logarithmic response cmos image sensor with on-chip calibration,” *IEEE Journal of Solid-state circuits*, vol. 35, no. 8, pp. 1146–1152, 2000.

[59] J. E. Garcia, A. G. Dyer, A. D. Greentree, G. Spring, and P. A. Wilksch, “Linearisation of rgb camera responses for quantitative image analysis of visible and uv photography: a comparison of two techniques,” *PLoS One*, vol. 8, no. 11, p. e79534, 2013.

[60] A. Glukhovsky, G. Meron, and G. J. Iddan, “Device and method for dark current noise temperature sensing in an imaging device,” Aug. 19 2003, US Patent 6,607,301.

[61] J. M. Fiske, “Flare light compensation,” Dec. 4 1990, US Patent 4,974,810.

[62] L. T. Maloney and B. A. Wandell, “Color constancy: a method for recovering surface spectral reflectance,” *JOSA A*, vol. 3, no. 1, pp. 29–33, 1986.

[63] G. Buchsbaum, “A spatial processor model for object colour perception,” *Journal of the Franklin institute*, vol. 310, no. 1, pp. 1–26, 1980.

[64] A. V. Durg and O. Rashkovskiy, “Global white point detection and white balance for color images,” May 30 2000, US Patent 6,069,972.

[65] E. H. Land and J. J. McCann, “Lightness and retinex theory,” *Josa*, vol. 61, no. 1, pp. 1–11, 1971.

[66] A. C. Hurlbert, “The computation of color,” 1989.

[67] Z.-u. Rahman, D. J. Jobson, and G. A. Woodell, “Multi-scale retinex for color image enhancement,” in *Proceedings of 3rd IEEE International Conference on Image Processing*, vol. 3. IEEE, 1996, pp. 1003–1006.

[68] V. Agarwal, B. R. Abidi, A. Koschan, and M. A. Abidi, “An overview of color constancy algorithms,” *Journal of Pattern Recognition Research*, vol. 1, no. 1, pp. 42–54, 2006.

[69] S. D. Hordley and G. D. Finlayson, “Reevaluation of color constancy algorithm performance,” *JOSA A*, vol. 23, no. 5, pp. 1008–1020, 2006.

[70] M. Ebner, *Color constancy*. John Wiley & Sons, 2007, vol. 6.

[71] A. Gijsenij, T. Gevers, and J. Van De Weijer, “Computational color constancy: Survey and experiments,” *IEEE Transactions on Image Processing*, vol. 20, no. 9, pp. 2475–2489, 2011.

[72] G. D. Finlayson, “Corrected-moment illuminant estimation,” in *The IEEE International Conference on Computer Vision (ICCV)*, December 2013.

[73] M. I. Guarnera, G. Messina, and V. Tomaselli, “Method and system for demosaicing artifact removal,” Apr. 26 2006, EU Patent 1650979A1.

[74] W. H. McIlhagga and K. A. May, “Optimal edge filters explain human blur detection,” *Journal of Vision*, vol. 12, no. 10, pp. 9–9, 2012.

[75] P. Satgunam, R. L. Woods, G. Luo, P. M. Bronstad, Z. Reynolds, C. Ramachandra, B. W. Mel, and E. Peli, “Effects of contour enhancement on low-vision preference and visual search,” *Optometry and vision science: official publication of the American Academy of Optometry*, vol. 89, no. 9, p. E1364, 2012.

[76] G. Deng, “A generalized unsharp masking algorithm,” *IEEE transactions on Image Processing*, vol. 20, no. 5, pp. 1249–1261, 2011.

[77] R. Mantiuk and H.-P. Seidel, “Modeling a generic tone-mapping operator,” in *Computer Graphics Forum*, vol. 27, no. 2. Wiley Online Library, 2008, pp. 699–708.

[78] G. Eilertsen, J. Unger, and R. K. Mantiuk, “Evaluation of tone mapping operators for hdr video,” in *High Dynamic Range Video*. Elsevier, 2016, pp. 185–207.

[79] G. K. Wallace, “The jpeg still picture compression standard,” *IEEE transactions on consumer electronics*, vol. 38, no. 1, pp. xviii–xxxiv, 1992.

[80] “Exchangeable image file format for digital still cameras: Exif Version 2.2,” Japan Electronics and Information Technology Industries Association, Japan, Standard, Apr. 2002.

[81] “Camera | android developers,” https://developer.android.com/reference/android/hardware/Camera, (Accessed on 04/05/2019).

[82] “Capturing photos in raw format | apple developer documentation,” https://developer.apple.com/documentation/avfoundation/cameras_and_media_capture/capturing_still_and_live_photos/capturing_photos_in_raw_format, (Accessed on 04/05/2019).

[83] “Adobe Digital Negative (DNG) Specification,” Adobe Systems Incorporated, 345 Park Avenue, San Jose, CA 95110-2704, United States, Tech. Rep., Jun. 2012.

[84] “Dngcreator | android developers,” https://developer.android.com/reference/android/hardware/camera2/DngCreator, (Accessed on 04/05/2019).

[85] F. Fang, H. Gong, M. Mackiewicz, and G. Finlayson, “Colour correction toolbox,” *AIC 2017 Jeju*, pp. 132–132, 2017.

[86] M. Anderson, R. Motta, S. Chandrasekar, and M. Stokes, “Proposal for a standard default color space for the internet - sRGB,” in *Color and imaging conference*, vol. 1996, no. 1. Society for Imaging Science and Technology, 1996, pp. 238–245.

[87] I. X-Rite, “ColorChecker Digital SG: X-Rite Photo & Video.” [Online]. Available: http://www.uef.fi/web/spectral/munsell-colors-matt-spectrofotometer-measured

[88] J. Jiang, D. Liu, J. Gu, and S. Süsstrunk, “What is the space of spectral sensitivity functions for digital color cameras?” in *Applications of Computer Vision (WACV), IEEE Workshop on*. IEEE, 2013, pp. 168–179.

[89] M. M. Darrodi, G. Finlayson, T. Goodman, and M. Mackiewicz, “Reference data set for camera spectral sensitivity estimation,” *JOSA A*, vol. 32, no. 3, pp. 381–391, 2015.

[90] University of Eastern Finland, “Spectral database - Agfa IT8.7/2 set.” [Online]. Available: https://www.uef.fi/en/web/spectral/agfa-it8.7/2-set

[91] “Graphic Technology - Color Reflection Target For Input Scanner Calibration,” American National Standards Institute, Reston, VA, US, Standard, Feb. 1993.

[92] S. Westland, J. Shaw, and H. Owens, “Colour statistics of natural and man-made surfaces,” *Sensor Review*, 2000.

[93] University of Eastern Finland, “Spectral database - munsell colors matt (spectrofotometer measured).” [Online]. Available: http://www.uef.fi/web/spectral/munsell-colors-matt-spectrofotometer-measured

[94] i. B. Munsell Color Company, *Munsell book of color : matte finish collection*. Baltimore : Munsell Color, 1976., 1976, S̈tandard method of specifying color by the Munsell system.,̧ ASTM D 1535 - 80, 23 p. in pocket.;S̈tandard method for visual evaluation of color differences of opaque materials.,̧ ASTM D 1729 - 69 (Reapproved 1974), 6 p. in pocket;M̈unsell neutral value scale, matte, 32-step scale, 1970 edition.;M̈unsell color standards–matte finish papers.: [Online]. Available: https://search.library.wisc.edu/catalog/999526993902121

[95] C. S. McCamy, H. Marcus, and J. Davidson, “A color-rendition chart,” *J. App. Photog. Eng*, vol. 2, no. 3, pp. 95–99, 1976.

[96] M. J. Vrhel, R. Gershon, and L. S. Iwan, “Measurement and analysis of object reflectance spectra,” *Color Research & Application*, vol. 19, no. 1, pp. 4–9, 1994.

[97] K. Barnard, L. Martin, B. Funt, and A. Coath, “A data set for color research,” *Color Research & Application*, vol. 27, no. 3, pp. 147–151, 2002.

[98] D. B. Judd, D. L. MacAdam, G. Wyszecki, H. Budde, H. Condit, S. Henderson, and J. Simonds, “Spectral distribution of typical daylight as a function of correlated color temperature,” *Josa*, vol. 54, no. 8, pp. 1031–1040, 1964.

[99] “Colorimetry – Part 2: CIE standard illuminants,” International Organization for Standardization, Geneva, CH, Standard, Oct. 2007.

[100] F. N. Fritsch and R. E. Carlson, “Monotone piecewise cubic interpolation,” *SIAM Journal on Numerical Analysis*, vol. 17, no. 2, pp. 238–246, 1980.

[101] P. V. Gehler, C. Rother, A. Blake, T. Minka, and T. Sharp, “Bayesian color constancy revisited,” in *2008 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2008, pp. 1–8.

[102] S.-B. Gao, K.-F. Yang, C.-Y. Li, and Y.-J. Li, “Color constancy using double-opponency,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 37, no. 10, pp. 1973–1985, 2015.

[103] A. Chakrabarti, “Color constancy by learning to predict chromaticity from luminance,” *arXiv preprint arXiv:1506.02167*, 2015.

[104] E. Garcia and M. Gupta, “Building accurate and smooth ICC profiles by lattice regression,” in *Color and Imaging Conference*, vol. 2009, no. 1. Society for Imaging Science and Technology, 2009, pp. 101–106.

[105] T. L. V. Cheung and S. Westland, “Color camera characterisation using artificial neural networks,” in *Color and Imaging Conference*, vol. 2002, no. 1. Society for Imaging Science and Technology, 2002, pp. 117–120.

[106] G. D. Finlayson, M. Mackiewicz, and A. Hurlbert, "Color correction using root-polynomial regression," *IEEE Transactions on Image Processing*, vol. 24, no. 5, pp. 1460–1470, 2015.

[107] M. Mackiewicz, C. F. Andersen, and G. D. Finlayson, "Method for hue plane preserving color correction," *J. Opt. Soc. Amer. A*, vol. 33, no. 11, pp. 2166–2177, 2016.

[108] B. Funt and P. Bastani, "Intensity independent rgb-to-xyz colour camera calibration," in *AIC (International Colour Association) Conference*, vol. 5, 2012.

[109] G. Finlayson, H. Gong, and R. Fisher, "Color homography color correction," *arXiv preprint arXiv:1607.05947*, 2016.

[110] G. D. Finlayson, M. Mohammadzadeh Darrodi, and M. Mackiewicz, "The alternating least squares technique for nonuniform intensity color correction," *Color Research & Application*, vol. 40, no. 3, pp. 232–242, 2015.

[111] M. J. Vrhel and H. Trussell, "Color correction using principal components," *Color Research & Application*, vol. 17, no. 5, pp. 328–338, 1992.

[112] G. D. Finlayson and M. S. Drew, "The maximum ignorance assumption with positivity," in *Color and Imaging Conference*, vol. 1996, no. 1. Society for Imaging Science and Technology, 1996, pp. 202–205.

[113] E. L. Krinov, "Spectral reflectance properties of natural formations," *National Research Council of Canada, Ottawa, 1947*, 1947.

[114] R. W. G. Hunt and M. R. Pointer, *Measuring colour*. John Wiley & Sons, 2011.

[115] "CIE 15: Technical Report: Colorimetry, 3rd edition," Commission internationale de l'éclairage, Vienna AUSTRIA, Tech. Rep., 2004.

[116] C. B. Barber, D. P. Dobkin, D. P. Dobkin, and H. Huhdanpaa, "The quickhull algorithm for convex hulls," *ACM Transactions on Mathematical Software (TOMS)*, vol. 22, no. 4, pp. 469–483, 1996.

[117] P. L. Vora and H. J. Trussell, "Measure of goodness of a set of color-scanning filters," *JOSA A*, vol. 10, no. 7, pp. 1499–1508, 1993.

[118] L. T. Maloney, "Evaluation of linear models of surface spectral reflectance with small numbers of parameters," *J. Opt. Soc. Amer. A*, vol. 3, no. 10, pp. 1673–1683, 1986.

[119] L. Van Der Maaten, E. Postma, and J. Van den Herik, "Dimensionality reduction: a comparative," *J Mach Learn Res*, vol. 10, no. 66-71, p. 13, 2009.

[120] G. D. Finlayson, Y. Zhu, and H. Gong, "Using a simple colour pre-filter to make cameras more colorimetric," in *Color and Imaging Conference*, vol. 2018, no. 1. Society for Imaging Science and Technology, 2018, pp. 182–186.

[121] P. L. Vora and H. J. Trussell, "Mathematical methods for the design of color scanning filters," *IEEE Transactions on Image Processing*, vol. 6, no. 2, pp. 312–320, 1997.

[122] J. Cohen, “Dependency of the spectral reflectance curves of the munsell color chips,” *Psychonomic Science*, vol. 1, no. 1-12, pp. 369–370, 1964.

[123] J. P. Parkkinen, J. Hallikainen, and T. Jaaskelainen, “Characteristic spectra of munsell colors,” *JOSA A*, vol. 6, no. 2, pp. 318–322, 1989.

[124] D. H. Marimont and B. A. Wandell, “Linear models of surface and illuminant spectra,” *JOSA A*, vol. 9, no. 11, pp. 1905–1913, 1992.

[125] A. R. Robertson, “The cie 1976 color-difference formulae,” *Color Research & Application*, vol. 2, no. 1, pp. 7–11, 1977.

[126] G. D. Finlayson, J. Vazquez-Corral, and F. Fang, “The discrete cosine maximum ignorance assumption,” in *Color and Imaging Conference*, vol. 2021. Society for Imaging Science and Technology, 2021.

[127] M. S. Drew and B. V. Funt, “Natural metamers,” *CVGIP: Image Understanding*, vol. 56, no. 2, pp. 139–151, 1992.

[128] G. D. Finlayson and P. Morovic, “Metamer sets,” *JOSA A*, vol. 22, no. 5, pp. 810–819, 2005.

[129] R. S. Berns, *Billmeyer and Saltzman’s principles of color technology*. John Wiley & Sons, 2019.

[130] M. E. Dyer, Z. Füredi, and C. McDiarmid, “Volumes spanned by random points in the hypercube,” *Random Structures & Algorithms*, vol. 3, no. 1, pp. 91–106, 1992.

[131] T. Smith and J. Guild, “The cie colorimetric standards and their use,” *Transactions of the Optical Society*, vol. 33, no. 3, p. 73, 1931.

[132] C. F. Van Loan and G. H. Golub, *Matrix computations*. Johns Hopkins University Press Baltimore, 1983.

[133] J. Vazquez-Corral, D. Connah, and M. Bertalmío, “Perceptual color characterization of cameras,” *Sensors*, vol. 14, no. 12, pp. 23 205–23 229, 2014.

[134] G. D. Finlayson and M. S. Drew, “White-point preserving color correction,” in *Color and Imaging Conference*, vol. 1997, no. 1. Society for Imaging Science and Technology, 1997, pp. 258–261.

[135] S. D. Hordley, J. F. Paul, and G. D. Finlayson, “Colour correction in theory and practice,” in *Visualization, Imaging, And Image Processing: Fifth IASTED International Conference Proceedings*, 2005.

[136] S. M. Newhall, D. Nickerson, and D. B. Judd, “Final report of the osa subcommittee on the spacing of the munsell colors,” *josa*, vol. 33, no. 7, pp. 385–418, 1943.

[137] M. S. Peercy, “Linear color representations for full speed spectral rendering,” in *Proceedings of the 20th annual conference on Computer graphics and interactive techniques*, 1993, pp. 191–198.

[138] G. D. Finlayson, M. S. Drew, and B. V. Funt, "Color constancy: generalized diagonal transforms suffice," *JOSA A*, vol. 11, no. 11, pp. 3011–3019, 1994.

[139] J. E. Gentle, *Numerical linear algebra for applications in statistics*. Springer Science & Business Media, 2012.

[140] N. Ahmed, T. Natarajan, and K. R. Rao, "Discrete cosine transform," *IEEE transactions on Computers*, vol. 100, no. 1, pp. 90–93, 1974.

[141] K. Fukuda, "Frequently asked questions in polyhedral computation," http://www.ifor.math.ethz.ch/ fukuda/polyfaq/polyfaq.html, 2004.

[142] B. Mulgrew and C. F. Cowan, *Adaptive filters and equalisers*. Springer Science & Business Media, 2012, vol. 56.

[143] E. Kreyszig, "Advanced engineering mathematics, 10th eddition," 2009.

[144] T. M. Cover, *Elements of information theory*. John Wiley & Sons, 1999.

[145] F. Fang and G. Finlayson, "Maximum ignorance polynomial colour correction," *AIC 2017 Jeju*, pp. 133–133, 2017.

[146] G. Sharma and H. J. Trussell, "Digital color imaging," *IEEE transactions on image processing*, vol. 6, no. 7, pp. 901–932, 1997.