Estimating Varying Illuminant Colours in Images

Stuart Ellis Lynch

A thesis submitted for the degree of
Doctor of Philosophy
at the University of East Anglia
March 2014

Estimating Varying Illuminant Colours in Images

Stuart Ellis Lynch

Acknowledgements

It's very surreal after four years of working on a project to see it finished and to think about all of those people who have helped me along the way. Of course, the first to start with is my supervisor Graham Finlayson. He has been a tough cookie at times, but in heinsight that was very much the kind of supervision I needed. His demand for excellence and his belief in his research group made it an honour to work in the group. I would also like to thank Mark Drew from Simon Fraser University in Canada. He had very much been in the background throughout the entirety of my PhD, to give advice and tell interesting stories. I also got the pleasure of working directly with him on a project, which eventually became the final chapter of my thesis. Also, thank you to Michael Brown who travelled all the way from Singapore to examine me in my viva.

I'd like to particularly thank Michael Harris for always being on hand to make sure my logic and rational was sound. He has been a great lab partner and an even better friend. Also to the other people who have been in the group along the way (Roberto, Michal, Perla, Dom, Chris, Toby and Maryam).

I would like to thank Apple Inc for giving me two incredible internships, and the oppurtunity to apply some of what I had learned in an industry environment (specifically my managers Jeremy Holland and Kjell Bronder). On a similar vain, thanks to Mixcloud (particularly Mat Clayton) for giving me a flexible job, which allowed me to eat in my last year when the funding had run out!

To complete, a few personal thank yous. My brother-in-law Mark Duckmanton, when I got accepted onto a PhD you gave your Nikon D60 SLR as a congratulations gift. I used this camera to collect all my data in the thesis, it was very much appreciated. To Stacey Williams, for always taking a supporting interest in everything I

did, and being on hand when I needed to get things off my chest. To Liam Draycott, for keeping me company in the Coach Makers Arms when I was writing up and being on hand whenever I needed to clarify if something made sense. Finally to Mum and Dad, for your unwavering belief in me I thank you from the bottom of my heart. Words cannot express how important that was at times.

Publications

- Graham D Finlayson and Stuart E Lynch. "Revisiting surface colour estimation under varying illumination." *Color and Imaging Conference. Vol. 2010, San Antonio, Texas*. No. 1. Society for Imaging Science and Technology, 2010.
- Stuart E Lynch, Mark S Drew and Graham D Finlayson "Colour Constancy from Both Sides of the Shadow Edge.", *Color and Photometry in Computer Vision Workshop at the International Conference on Computer Vision, Sydney Australia*, 2013

Abstract

Colour Constancy is the ability to perceive colours independently of varying illumination colour. A human could tell that a white t-shirt was indeed white, even under the presence of blue or red illumination. These illuminant colours would actually make the reflectance colour of the t-shirt bluish or reddish. Humans can, to a good extent, see colours constantly. Getting a computer to achieve the same goal, with a high level of accuracy has proven problematic. Particularly if we wanted to use colour as a main cue in object recognition. If we trained a system on object colours under one illuminant and then tried to recognise the objects under another illuminant, the system would likely fail. Early colour constancy algorithms assumed that an image contains a single uniform illuminant. They would then attempt to estimate the colour of the illuminant to apply a single correction to the entire image.

It's not hard to imagine a scenario where a scene is lit by more than one illuminant. If we take the case of an outdoors scene on a typical summers day, we would see objects brightly lit by sunlight and others that are in shadow. The ambient light in shadows is known to be a different colour to that of direct sunlight (bluish and yellowish respectively). This means that there are at least two illuminant colours to be recovered in this scene. This thesis focuses on the harder case of recovering the illuminant colours when more than one are present in a scene.

Early work on this subject made the empirical observation that illuminant colours are actually very predictable compared to surface colours. Real-world illuminants tend not to be greens or purples, but rather blues, yellows and reds. We can think of an illuminant mapping as the function which takes a scene from some unknown illuminant to a known illuminant. We model this mapping as a simple multiplication of the Red, Green and Blue channels of a pixel. It turns out that the set of realistic

mappings approximately lies on a line segment in chromaticity space. We propose an algorithm that uses this knowledge and only requires two pixels of the same surface under two illuminants as input. We can then recover an estimate for the surface reflectance colour, and subsequently the two illuminants.

Additionally in this thesis, we propose a more robust algorithm that can use varying surface reflectance data in a scene. One of the most successful colour constancy algorithms, known Gamut Mappping, was developed by Forsyth (1990). He argued that the illuminant colour of a scene naturally constrains the surfaces colours that are possible to perceive. We couldn't perceive a very chromatic red under a deep blue illuminant. We introduce our multiple illuminant constraint in a Gamut Mapping context and are able to further improve it's performance.

The final piece of work proposes a method for detecting shadow-edges, so that we can automatically recover estimates for the illuminant colours in and out of shadow. We also formulate our illuminant estimation algorithm in a voting scheme, that probabilistically chooses an illuminant estimate on both sides of the shadow edge.

We test the performance of all our algorithms experimentally on well known datasets, as well as our new proposed shadow datasets.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

For decades the problem of getting a computer to see has proven notoriously problematic. An ideal vision system would be able to draw as much high-level information out of an image as possible. For example, in Figure 1.1, we can deduce that it is daytime and that there are people in the scene. The lady in the middle, who is the focus of the photograph is carrying shopping bags, which implies she was shopping. You might therefore suppose that she is in a city centre. The picture was probably taken somewhere in the UK because of the architecture and the shop names written on the shopping bags. No current Computer Vision system is able to interpret an image in this way.

Broadly speaking we can think of the colours we see in the world as a combination of surface reflectance colour and light colour. Light is shone onto a surface, the light is modulated by the surface and a new spectrum is reflected back (Figure 1.2). The signal that reaches our eyes therefore changes as the light colour and surface colours do. This presents a problem when trying to understand the colour of an object in a consistent way.

Colour constancy, refers to the mechanism through which we are reliably able to distinguish and recognise objects of different colours, independent of the light. Without this, different recognition methods that use colour, would only be valid

Figure 1.1: Lady with shopping.

under fixed illumination conditions. In Figure 1.3, left we see a colour image of fruit and right the monochrome version. The importance of colour in determining ripeness is clear.

Figure 1.4 illustrates the ripeness problem when illumination colour is put into the mix. The left panel shows two bunches of bananas. The left bunch is unripe, we know this because of it's greenish colour. The right bunch look more edible, because of their yellowish colour. However, the blue light makes the ripe yellow banana induce a green response. The change in illuminant has made ripeness harder to determine. Colour constancy algorithms attempt to infer and remove the light colour, and in so allow fruit ripeness to be correctly determined.

Colour constancy is useful in object detection and recognition, which is one of the most targeted features in Computer Vision (Abdel-Hakim and Farag, 2006; Funt et al., 1998; Funt and Finlayson, 1995; Gevers and Smeulders, 1999; Lowe, 1999; Swain and Ballard, 1991; Tsin et al., 2001). In this context colour constancy algorithms attempt

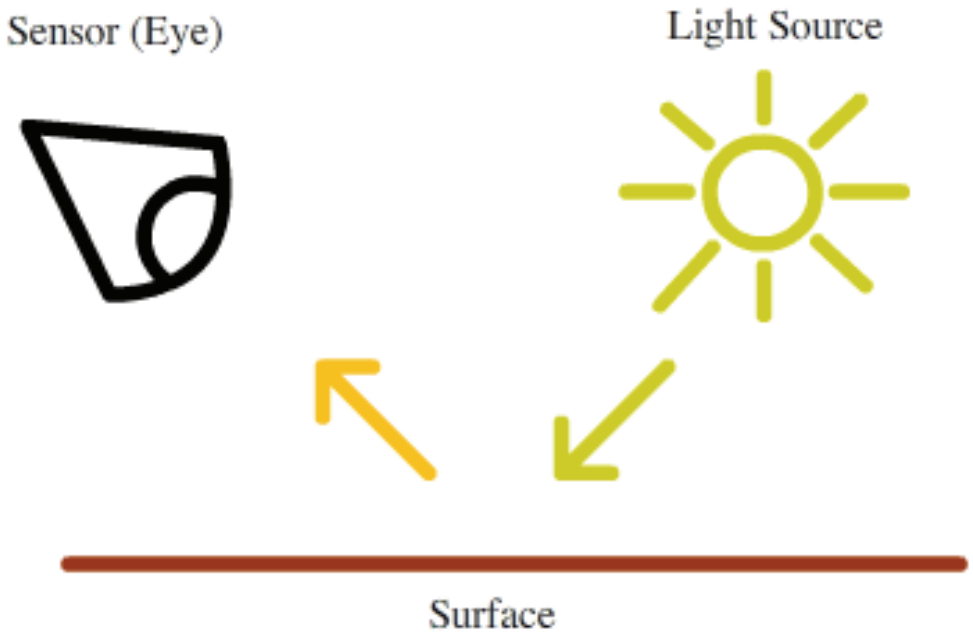

Figure 1.2: Light is shone onto a surface, the colour is changed and the new signal is reflected back.

to infer the colour cast due to illumination and then remove this colour cast from images. This has importance in photography: when we take pictures we wish the image to look correct.

Colour constancy algorithms can be split up into roughly 5 categories: low-level statistics, neural networks, gamut mapping, probabilistic and physics-based. Low-level statistical based methods make simple assumptions about the statistics of an image, and estimate the illuminant as an image statistic. Neural network based methods train a network on a large set of images in which the illuminant is known. Gamut mapping methods make assumptions about the surface and illuminant colours that are likely to exist in the world. The reddest red cannot occur under the bluest light Forsyth (1990). Probabilistic methods attempt to collect the likelihood of a given pixel being observed under a given illuminant using a statistical prior. Finally, physics

Figure 1.3: On the left is a colour image of a ripe tomato among some unripe ones. The right is a 'brightness' (monochromatic) image of the same scene. In the left image it is easy to determine which tomato is ripe, it is not so easy in the right.

based methods reason about how the image is formed, and use that information to draw conclusions about the illuminant (e.g. highlights in images are a good cue for illuminant colour). Each has their strengths and weaknesses. There is no one algorithm that out-performs all the others all of the time. In this vain, it has become common to develop algorithms that work well in isolated cases. Such as in images that exhibit certain statistical characteristics (Gijsenij and Gevers, 2007) or in images that contain faces (Bianco and Schettini, 2012).

Accuracy in illuminant colour estimation is often relative to image content. Images which have high surface colour variation and have a single illuminant colour are often easier to correct automatically. Forsyth (1990) showed that increasing surface colour variation provides a natural constraint to the feasible set of scene illuminants. Conversely, scenes which have little surface colour variation and/or have more than one illuminant are hard cases for traditional colour constancy algorithms. It is not hard to conceptualise a scene that fits into either of these latter categories. A photograph of yellow flowers in a field or of a green forest could have limited variation in

Figure 1.4: Image of ripe and unripe bananas. The right image has a bluish colour filter applied such that yellow bananas in the right image have the same pixel values as the green bananas in the left.

colour. Scenes which have different light sources and contain shadows have multiple illuminants. If we imagine a scene on a typical summer day. The objects in the scene that occlude the suns light would cast a shadow onto the ground. However, the areas in shadow are still visible, because not all light is blocked. The remaining light is ambient and bluish. This is in contrast to the yellowish sunlight. In Figure 1.5 we show 2 example images from a dataset we discuss in Chapter 3. Roughly half of each image is in shadow. The measured illuminant colour is shown below each image. Even though both measurements are taken in the same scene, we see a substantially different measurement. The white patch on the colour chart in the left image shows a reddish illuminant and the one on the right is blueish.

This very plausible scenario could wreak havoc with an object recognition system that is tuned to recognise colours - such as the system proposed by Swain and Ballard (1991). The palette of surface reflectance colours will shift in two different directions (red and blue). A preprocessing step which could understand illuminant colours in a scene would improve performance of this process.

Figure 1.5: Colour chart placed in and out of shadow. **Left:** Illuminant measured out of shadow **Right:** Illuminant measured in shadow.

For camera white balance the goal is to render an image such that the illuminant colour cast is normalised. If we take an example image which contains shadows where we know the illuminant colours inside and outside of shadow, we can 'correct' the image in two ways. Traditionally an automatic white balance algorithm would apply a global correction to every pixel in the scene. Figure 1.6 shows an image captured by a camera on the left. To the right we correct the image with the illuminant measured in shadow, and below it the illuminant out of shadow. In nearly all colour constancy datasets, the illuminant is measured by arbitrarily placing a colour checker inside or outside of shadow. That measurement is then used for correction. In the case shown here it is not clear what the correct procedure is, as both 'correct' answers give significantly different results.

In this thesis we do not focus on how we white balance an image with multiple illuminants, but rather on the actual estimation of multiple illuminant colours in a single scene (with a focus on shadows). Comparatively sparse literature has examined

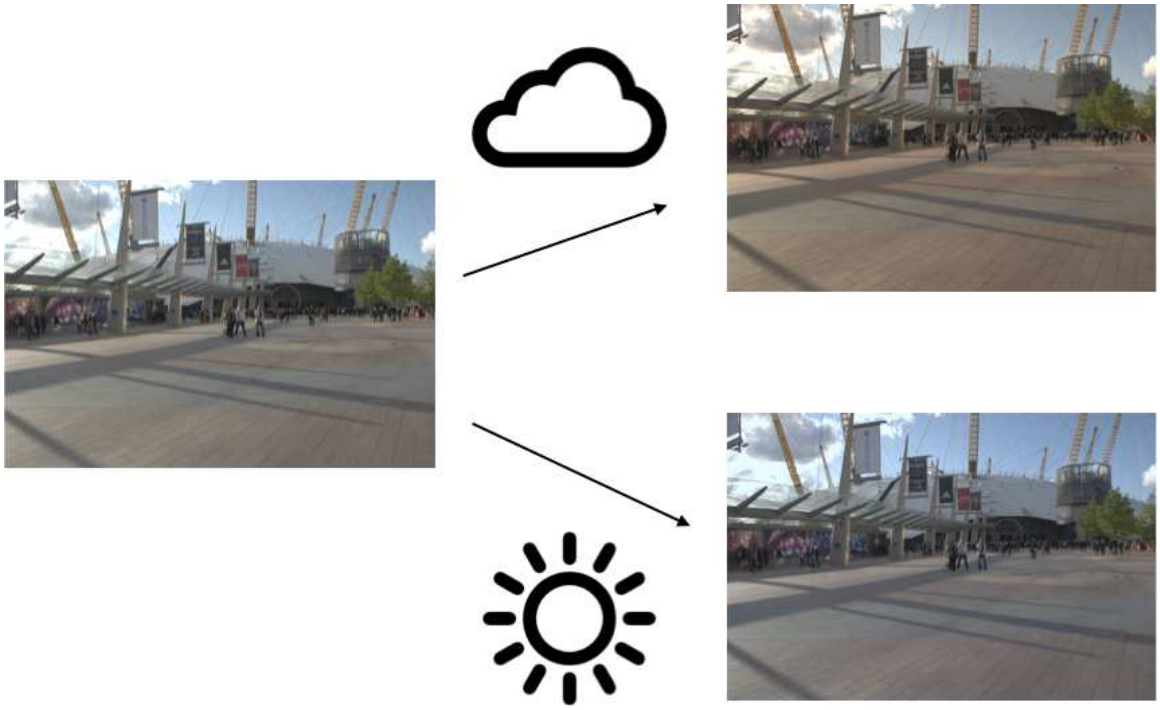

Figure 1.6: **Left:** An image containing shadows (no white balancing). **Right:** The same image corrected to the illuminants measured inside and out of shadow.

this problem however some works have attempted to tackle it (Finlayson et al., 1995; Gijsenij et al., 2012; M Bleier et al., 2011). It seems that a level of segmentation is unavoidable. If we wanted to completely remove the affects of illumination (like in the work of Finlayson et al. (2002a) on shadow removal) we would need to know where the areas of different illumination in an image are. The colours of illumination that we are likely to perceive in the world are a lot more constrained than colours of surfaces. Lights are typically red, yellow white or blue (Finlayson, 1996). We are interested in using this property directly to be able to estimate the illuminant colours in a scene, as well as identify areas of changing illumination (i.e. shadow edges).

1.1 Background

The process in which an image is formed is complex. A ray of light can under-go all sorts of changes before it reaches the eye. Surface properties affect how the light is reflected and how the spectrum is altered. In the simplest case, we can model image formation in a world of uniformly lit, frontally facing, matte surfaces. This means that the spectrum of light is the same across the scene, and it is shone directly onto the surface. The spectrum of the light is modified by the surface and reflected back as a different colour. We can model a spectrum of light as a function of wavelength λ which spans the visible spectrum ω (approximately between 400 and 700 μm) and we model this spectrum as $E(\lambda)$. We then define a function which represents the reflectance of the matte surfaces in a scene $S^x(\lambda)$. The function is dependent on the surface x, as surfaces can vary in the scene. The product of these two functions represents the light that is reflected from a surface. We define a function $R(\lambda, k)$ which represents the sensitives of k sensors ($k = R, G, B$). The integral of the product of all of these functions gives us our standard image formation equation at a pixel p_k^x (Forsyth, 1990; Land and Center, 1974; Maloney and Wandell, 1986):

$$p_k^x = \int_\omega S^x(\lambda)E(\lambda)R_k(\lambda)d\lambda, \tag{1.1}$$

where ω denotes the visible spectrum. In most cameras k represents 3 sensors that are sensitive in the Red, Green and Blue parts of the spectrum (referred to in future as RGB).

1.1.1 The Algebra of Image Formation

Equation 1.1 describes a model of image formation that is the integral of the product of three infinite functions. To simplify this many authors (Finlayson et al., 2002b; Forsyth, 1990; Weijer and Gevers, 2007) have opted to use a model of image formation

that is discrete and operates entirely on Red, Green and Blue values. If we model our sensors as functions that peak at a single wavelength (Dirac Delta functions) and whose integral is 1 then this equation simplifies. If the k^{th} sensor peaks at λ_k then the k^{th} sensor response is $\int_\omega S^x(\lambda_k)E(\lambda_k)R_k(\lambda_k)$ which is equal to $S^x(\lambda_k)E(\lambda_k)$. We can therefore model image formation at a pixel as a simple multiplication of light and surface RGBs. Since algebra doesn't provide a standard notation for point-wise vector operations we can write the illuminant RGBs as the diagonal elements of a matrix. This produces the following diagonal matrix transform which approximates image formation

$$\begin{bmatrix} p_R^x \\ p_G^x \\ p_B^x \end{bmatrix} \approx \begin{bmatrix} E(\lambda_R) & 0 & 0 \\ 0 & E(\lambda_G) & 0 \\ 0 & 0 & E(\lambda_B) \end{bmatrix} \begin{bmatrix} S^x(\lambda_R) \\ S^x(\lambda_G) \\ S^x(\lambda_B) \end{bmatrix}, \tag{1.2}$$

where λ_R, λ_G and λ_B are wavelengths in the Red, Green and Blue parts of the spectrum respectively. Clearly modelling sensors as Dirac Delta functions is a gross approximation, however if our sensors are sufficiently narrowband then this model holds for most typical sensors (Finlayson et al., 1994). We represent this equation algebraically as

$$\underline{p}^x \approx \boldsymbol{E}\underline{S}^x, \tag{1.3}$$

where $\boldsymbol{E}$ is a diagonal matrix with the k^{th} diagonal element $\int_\omega E(\lambda)R_k(\lambda)d\lambda$ and $\underline{S}^x$ is a vector with k^{th} element $\int_\omega S^x(\lambda)R_k(\lambda)d\lambda$. Our pixel RGB at surface x is therefore represented by the vector $\underline{p}^x$. Illuminant estimation can be thought of as finding $\boldsymbol{E}$ given only $\underline{p}^x$.

1.1.2 Two Illuminant Colour Constancy

In the scenario where our image contains two or more illuminants the function $E(\lambda)$ in Equation 1.3 would vary. This is normally assumed to be a constant. Let's assume that our goal was to individually correct every pixel in a scene, such that the illuminant was a uniform white (i.e. having the RGB value [1, 1, 1]). We are therefore interested in solving for a mapping which takes our pixel RGB $\underline{p}^x$ to an illuminant invariant representation of our surface colour $\underline{S}^x$. We can write this as the following

$$\begin{aligned} S_R^x &= p_R^x \times 1/E_R \\ S_G^x &= p_G^x \times 1/E_G \\ S_B^x &= p_B^x \times 1/E_B \end{aligned} \quad , \tag{1.4}$$

where E represents the illuminant RGB. We can think of $1/E$ as an illuminant mapping. This mapping takes a pixel from a state which is affected by illumination colour to a state which is not. For easier algebraic notation we can write the inverse of matrix $\boldsymbol{E}$ as $\boldsymbol{E}^{-1}$ where the three diagonal elements equal $1/E_R$, $1/E_G$ and $1/E_B$ respectively. Therefore we can rearrange Equation 1.3 to become

$$\underline{S}^x \approx \boldsymbol{E}^{-1}\underline{p}^x. \tag{1.5}$$

Finlayson (1996) argued that an entirely colour constant descriptor of $\underline{p}^x$ is unrealistic. This is because of the discrepancy in brightness between a bright surface illuminated by a dim illuminant and a dark surface illuminated by a bright illuminant. Both scenarios could produce exactly the same pixel response. He therefore proposed working in a space that is completely normalised to brightness variation. Colour brightness is ignored is called chromaticity. A typical representation of chromaticity is as follows

$$\begin{aligned} p_r &= p_R/(p_R + p_G + p_B) \\ p_g &= p_G/(p_R + p_G + p_B) \\ p_b &= p_B/(p_R + p_G + p_B) \end{aligned}, \tag{1.6}$$

where the lowercase representation of rgb denote chromaticity coordinates. It is trivial to show that the RGB vectors $\underline{p}$ and $\alpha\underline{p}$ have the same chromaticity, where α is some arbitrary scalar. Unfortunately, if we were to just substitute chromaticites into Equations 1.3 and 1.5, they would no longer be accurate. Finlayson (1996) therefore proposed using the following spectral-band ratio chromaticities

$$\begin{aligned} c_{r/k} &= p_R/p_k \\ c_{g/k} &= p_G/p_k \\ c_{b/k} &= p_B/p_k \end{aligned}, \tag{1.7}$$

where $k \in \{R, G, B\}$. If we choose k to equal Blue, then the spectral-band chromaticites equal $[p_R/p_B, p_G/p_B, 1]$. It is clear that chromaticites are 2-dimensional because the third coordinate is always 1. Therefore, it is common to drop the 1 in the notation. The benefit of using this representation is that we can substitute chromaticites directly into Equations 1.3 and 1.5, and they will still be correct. In expanded matrix form

$$\begin{bmatrix} c_{r/b} \\ c_{g/b} \end{bmatrix} \approx \begin{bmatrix} E_R/E_B & 0 \\ 0 & E_G/E_B \end{bmatrix} \begin{bmatrix} S_R/S_B \\ S_G/S_B \end{bmatrix}. \tag{1.8}$$

Finlayson et al. (1995) were interested in the set of plausible values that $\boldsymbol{E}$ could take. Illumination colours in the world are constrained. In the real-world we don't come across illumination colours that vary with all the colours of the rainbow. Real-world illuminant colours tend to be Reds, Yellows, Whites and Blues. Asserting that $\boldsymbol{E}$ falls within a restricted set of values therefore seems sensible. Illumination mappings $\boldsymbol{E}^{-1}$ were of particular interest because of an important property exhibited

by them in the real-world. Illumination mapping chromaticities (in spectral-band chromaticity space), have approximately one dimensional variance. This means that they can be approximated by a line, as shown in Figure 1.7 which plots real-world illuminant mappings in chromaticity space.

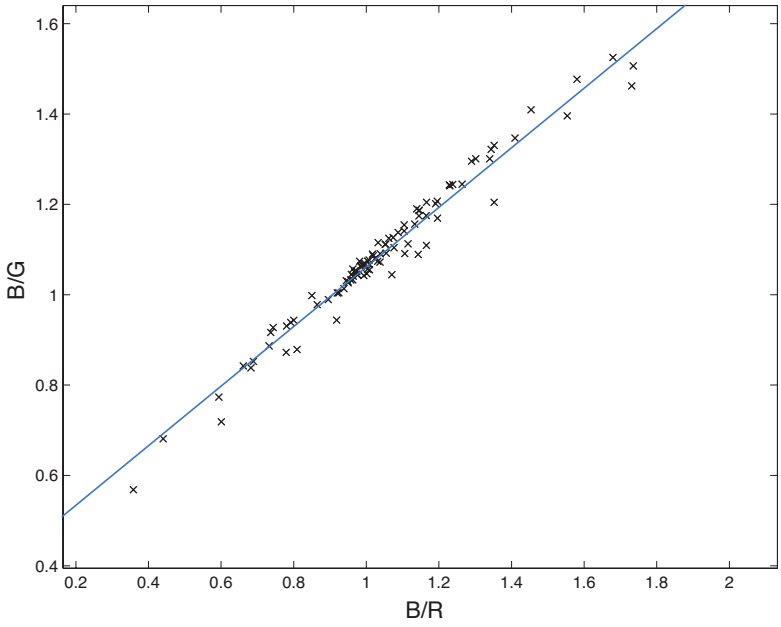

Figure 1.7: Illuminant Spectra captured by Hernández-Andrés et al. (2001), plotted in spectral band ratio chromaticity space.

This remarkable result, provides a very powerful constraint for illuminant estimation. Finlayson et al. (1995) made the assumption that all real-world illuminant mappings lie exactly on a line, and further assumed that this line can be predetermined. Following this they showed that it is possible to solve for $\underline{S}^x$ given two pixels of the same surface under two lights. This condition on pixels can be found in real images at a shadow edge. Imagine a red t-shirt with half of it in shadow. We can take pixels of the t-shirt from both sides of the shadow edge, and according to Finlayson

et al. (1995) that is all that is required to estimate the surface reflectance chromaticity, and consequently the two illuminant chromaticities. If the two input illuminants were substantially different (i.e. red and blue) the algorithm performed best. The method was shown not to be robust in real-world scenarios (Kawakami and Ikeuchi, 2009; Kawakami et al., 2005), but performed well in controlled conditions. The work of this thesis builds on this foundation by proposing illuminant estimation algorithms that have a real-world application in images that contain two (or more) illuminants.

1.2 Evaluating Illuminant Estimation

To understand how good an illuminant estimate is, a measure of error is required. As we cannot realistically expect to recover an accurate measure of illumination intensity, our error measure should account for this. For this reason, many authors have opted to use the angular distance between the RGB estimate and the actual RGB of the illuminant. The correct illuminant RGB is typically measured by placing a white patch in a scene (i.e. a patch that has spectrally uniform reflectance in the visible spectrum). The RGB pixels captured in this region of the image accurately represent the RGB of the illuminant. When testing an algorithm on the image, the white patch could be cropped out to avoid bias. We can think of an RGB as a vector in 3-dimensional space. The following equation defines the angle between two vectors $\underline{A}$ and $\underline{B}$

$$\theta = \cos^{-1}\left(\frac{\underline{A}\cdot\underline{B}}{||\underline{A}||\ ||\underline{B}||}\right), \tag{1.9}$$

where the symbol $||\ ||$ represents the vector's magnitude. In Figure 1.8 we show some examples of angular distances away from some typical illuminants. We can see that a difference of 12 degrees can be the difference between a bluish and a reddish colour. Visually, an acceptable threshold needs to be less than 5 degrees angular distance.

Angles with an error larger than this can have a noticeable change in colour. Many authors use the median angular error as a basis for algorithm comparison (Hordley and Finlayson, 2004).

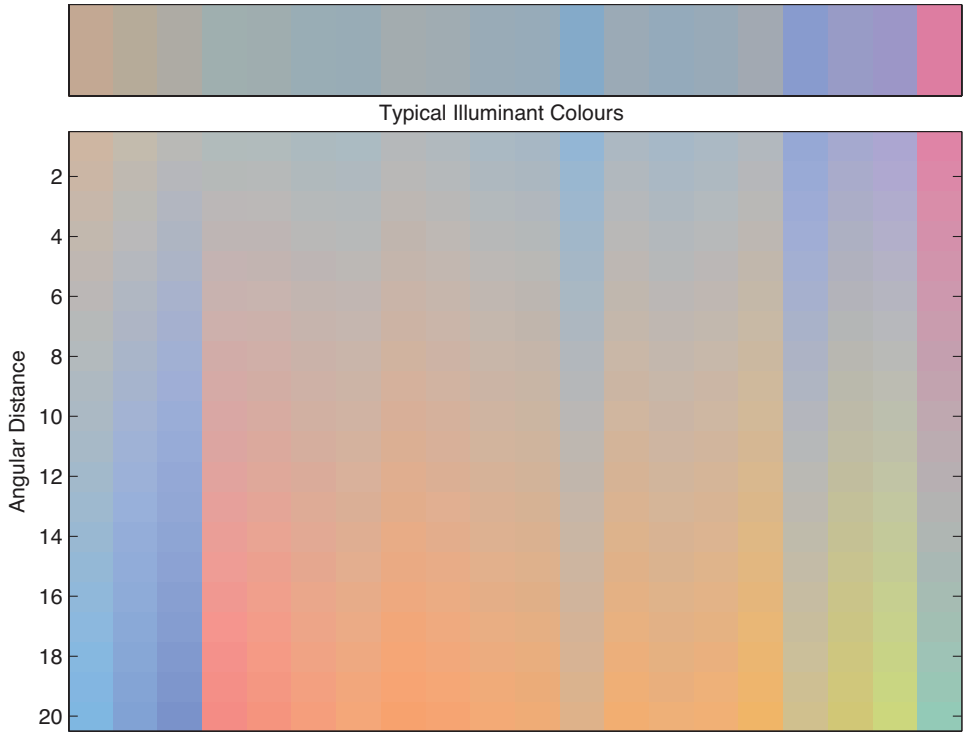

Figure 1.8: Typical illuminant colours shown with examples of different angular distances from the original; ranging from 0 to 20 degrees.

1.3 About This Thesis

We have explained that Colour Constancy is typically approached as a problem which tries to recover a single illuminant RGB given the sensor responses. There is now a large breadth of literature which approaches the problem in this way. We review some

of these in Chapter 2. However comparatively small literature focuses on scenes which do not fit the single illuminant assumption. One of the most common scenarios where we see more than one illuminant in a scene, is where shadows are present. Ambient daylight and light directly from the sun vary significantly in colour. The way that light interacts with the atmosphere also means that these colours change throughout the day. Therefore trying to associate a single uniform illuminant colour for every pixel in the scene could be the incorrect thing to do. This thesis focuses on these cases specifically, and develops several algorithms which attempt to use the varying illumination information directly to derive a solution for the illuminant colours in the scene.

1.3.1 Main Contributions

In this work we focus on the problem of multiple illuminant estimation in scenes. We list the main contributions of this thesis below:

- We provide two new datasets that contain images with shadows (i.e. two illuminants). For each image two illuminant RGBs are provided (Chapter 3).
- Additionally, we discovered that the popular Shi and Funt (2009) linear rendering of the Gehler et al. (2008) dataset had not been rendered as stated. We have provided a corrected linear rendering of this dataset.
- An adjustment was made to the methodology of a previous two illuminant estimation algorithm (Chapter 4). Our method proposes a method for finding a point simultaneously closest to two line segments. The work shows a large increase in performance on a common colour constancy dataset.
- We improved the work of Kawakami and Ikeuchi (2009) by adding an optimised calibration step to their algorithm.

- In Chapter 5 we incorporate the line segment technique into a Gamut Mapping (Forsyth, 1990) framework. The algorithm proposed takes multiple illuminants and surfaces and outputs estimates for the scene illuminants.

- To formulate our work into a complete illuminant estimation algorithm we propose a method for identifying shadows edges in images (Chapter 6). This method uses the Mean-Shift algorithm (Comaniciu and Meer, 2002) for segmentation.

- Finally, to complement our shadow edge detection, we propose an illuminant estimation algorithm that uses a voting procedure. This makes it more robust to input errors.

Chapter 2

Illuminant Estimation & Colour Constancy

Colour Constancy algorithms, are often approached as a process which takes an image as input and outputs an estimate for the illuminant RGB, which is then used to apply a single correction to every pixel in the image. To assess performance, datasets are built where the illuminant is known. The illuminant RGB is measured by placing a white patch in the scene. Because grey reflects all the wavelengths of the visible spectrum equally the RGB for the grey patch should correlate with the illuminant colour.

In this Chapter we shall review several common classes of traditional illuminant estimation algorithms. These algorithms mostly make the assumption that an image contains pixels of surfaces under a single uniform illuminant. We also review the algorithms which specifically focus on varying illumination (the focus of this thesis).

2.1 Overview

Hordley (2006) reviewed many of the well known colour constancy algorithms and puts them into 5 categories. They were: simple statistical methods, neural networks, gamut mapping, probabilistic methods and physics-based methods. He discussed the ways in which algorithms should be compared to each other in order to gage their

relative performance. He concluded that this task was not trivial as algorithm's performance is very much image dependent. It also is application dependent as to whether an algorithm has "good" performance. He referenced several studies which investigated whether current colour constancy algorithms would be good enough to enable colour-based object recognition. The answer then was a resounding no. This is arguably still the case today.

Gijsenij et al. (2011) recently did a comprehensive review of many popular colour constancy algorithms, on several common datasets (more on datasets in the next chapter). All of the algorithms tested assume a variety of surface colours under a single uniform illuminant. This means that all pixels in an image can be used as input to an algorithm, and then processed in the same way. They highlight that shadows are often ignored for simplicity. However, shadows are inherently a change in illumination, and so violate the assumptions of most algorithms. Gijsenij and Gevers (2007) looked at analysing image statistics in order to to choose which algorithm to use. The authors stated that algorithm performance is dependent on the input image, and different algorithms can have pros in different circumstances. In this thesis we wish to consider the cases in a which a scene is lit by more than one light.

Other algorithms such as the work of Bianco and Schettini (2012) detect the specific cases that they can solve well. In this particular work, if an image contains faces and skin tones then illuminants are estimated using Gamut mapping. Faces are found using the framework developed by Viola and Jones (2001). Good results are found for images which contain faces. A grand colour constancy algorithm in future may consist of using many algorithms in different circumstances.

2.2 Simple Assumptions

In this section we look at a class of algorithm, which computes simple statistics from the pixel values, in order to infer the illuminant colour. These simple algorithms require no calibration and little computing power. However, their simplicity often limits their ability to make accurate estimations of the illuminant colour.

2.2.1 GreyWorld

Let's imagine a scene with a very broad range of surface colours where each pixel RGB can be modelled using equation 1.3. Let's also say that the pixels were captured under a uniform white illuminant. So essentially $\boldsymbol{E}$ is the identity matrix. Suppose that the mean of all of these pixels was somewhere close to $[0.5, 0.5, 0.5]$. If we think of this as a vector, it has the same direction as the scene illuminant [1,1,1]. If we change the illuminant of the scene by substituting a second light into equation 1.3, the mean will also change. However, the change in direction of the mean vector and the direction of the illuminant vector will remain the same. Put another way: the mean of the image returns the correct illuminant colour.

The observation that the mean colour correlates with the light colour is the basis of the GreyWorld hypothesis formalised by Buchsbaum (1980) (amongst others). The author proposed that given a broad enough range of surfaces in a scene the mean pixel RGB should have the same colour as the illuminant. The implication is that the mean pixel of an image should be grey, any deviation from that is due to illumination. This clearly does not hold in scenes which do not have a broad range of colours (such as a forest which is mostly greens and browns). Yet assuming that the illuminant is correlated with the mean can work well. Though when it fails, it fails badly.

2.2.2 MaxRGB

Following the work of Land and McCann (1971), a similar algorithm to GreyWorld was developed. Broadly speaking, if a scene contains a white surface, the pixel response of that surface is a good estimate for the illuminant RGB. This is because for it to appear white, very little (if any) of the illuminant energy is absorbed and it is all reflected evenly. As a consequence the pixels are likely to be among the brightest in the scene (because they reflect the most energy). Therefore calculating the maximum RGB is proposed as an algorithm for estimation of the scene illuminant. This is often called the MaxRGB algorithm. Significantly, MaxRGB works where there is a white patch.

Let's imagine an image containing patches of all different colours except white. The MaxRGB statistic could still be a good one because a yellow patch and a blue patch together could reflect the same as a white patch. However, it still falls prey to the scenes that are dominated by a small group of colours. Using MaxRGB on a leafy green scene, would probably output a greenish illuminant.

2.2.3 Shades of Grey

Finlayson and Trezzi (2004) investigated GreyWorld and MaxRGB algorithms and found that they are both instances of Minkowski norms:

$$||\underline{X}||_p = \frac{1}{N}\left\{\sum_{i=1}^{N}|X_i|^p\right\}^{\frac{1}{p}}, \tag{2.1}$$

where $|X_i|^p$ is the absolute value of X_i raised to the power of p. In the case of $p = 1$ then this formula becomes the mean average (GreyWorld). In the case of $p \to \infty$ then this equation returns the maximum value (MaxRGB). Experimentally, they found that a p norm of 6 performs better on average than MaxRGB and GreyWorld. However, like GreyWorld and MaxRGB, for scenes with little colour variance Shades

of Grey can perform poorly.

2.2.4 Grey-Edge

Weijer and Gevers (2007) proposed an algorithm based on the Minkowski norm approach. Based on first, second and higher order derivatives (edges). The formal definition of their approach is

$$e_k^{n,p,\sigma} = \left(\int \left| \left| \frac{\delta^n I_k^\sigma(x)}{\delta x^n} \right| \right|^p dx \right)^{1/p} \qquad k \in \{R, G, B\}, \tag{2.2}$$

where $I_k^\sigma(x)$ denotes the convolution of the image I_k with a Gaussian filter G, with σ pixel standard deviation and n is the order of the derivative. When $n = 0$ this means no derivative is taken and $n = 1$ is the first-order derivative. The output $e_k^{n,p,\sigma}$ is the estimated RGB of the illuminant, to an unknown scaling. The authors have reported using edge information in this way delivers a significant improvement in median angular error compared to GreyWorld and MaxRGB.

2.3 Finite Dimensional Linear Models

There has been several works (DZmura, 1992; Funt and Drew, 1988; Gershon et al., 1988) which have considered whether it would be possible to recover the spectrum of the scene illuminant, given only the tristimulus values of the sensory system. To see how spectral quantities might be recovered let us represent the illuminant spectra as a weighted sum of representative basis functions

$$E(\lambda) = \sum_{i=1}^{n} \epsilon_i e_i(\lambda), \tag{2.3}$$

where n is the number of basis functions and ϵ_i is the weight for the i^{th} basis function. Let us represent surface reflectance spectra in the same way.

$$S^x(\lambda) = \sum_{j=1}^{m} \sigma_i^x s_i(\lambda). \tag{2.4}$$

The basis functions for spectra are usually determined by Principal Component Analysis (Parkkinen et al., 1989), or Characteristic Vector Analysis (Maloney and Wandell, 1986). Substituting equations 2.3 and 2.4 into the image formation equation 1.1 we get

$$p_k^x = \int_\omega \sum_{i=1}^{n} \sum_{j=1}^{m} \epsilon_i \sigma_j^x [e_i(\lambda) s_j(\lambda)] R_k(\lambda) d\lambda, \tag{2.5}$$

Maloney and Wandell (1986) rewrite equation 2.5 as a lighting matrix, premultiplying a surface weight vector $\underline{\sigma}^x$

$$\underline{p}^x = \mathbf{\Lambda}_\epsilon \underline{\sigma}^x, \tag{2.6}$$

where the kj^{th} entry of $\mathbf{\Lambda}_\epsilon$ equals $\int_\omega \sum_{i=1}^{n} [\epsilon_i e_i(\lambda)] s_j(\lambda) R_k(\lambda) d\lambda$.

Let's examine the case of a tristimulus system where we have a 3-dimensional model of illumination and a 2-dimensional model surface reflectance. In this case $\underline{p}^x$ lies on a 2-dimensional plane in 3-dimensional space. This plane is determined by ϵ. Further, we know that the plane will go through the origin because $[0, 0, 0]$ will always be a plausible pixel response. Therefore we only need two additional points to define the plane. The vector $\underline{N}_\epsilon$ characterises the plane: $\underline{N}_\epsilon = \underline{p}^1 \times \underline{p}^2$, where $\times$ is the vector cross product.

Let's define a matrix $\mathcal{T}_i$ with jk^{th} element $\int_\omega s_i(\lambda) e_j(\lambda) R_k(\lambda) d\lambda$. $\mathcal{T}_i \underline{\epsilon}$ is the response of the i^{th} surface basis under light $\underline{\epsilon}$. Following this, $\underline{\epsilon}$ must satisfy

$$\underline{N}_\epsilon^\top \mathcal{T}_i \underline{\epsilon} = 0. \tag{2.7}$$

So $\mathcal{T}_i \underline{\epsilon}$ lies on the plane (defined by $\underline{\epsilon}$) it must be orthogonal to the plane normal. In this form we cannot solve for $\underline{\epsilon}$ because we are only representing a single surface

basis function. By including both basis functions then we can solve for a unique, non-zero solution for $\underline{\epsilon}$

$$\begin{bmatrix} N^\top \mathcal{T}_1 \\ N^\top \mathcal{T}_2 \end{bmatrix} \underline{\epsilon} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}. \tag{2.8}$$

Now that we know $\underline{\epsilon}$ we can then solve for $\underline{\sigma}^x$ as

$$\underline{\sigma}^x = [\mathbf{\Lambda}_\epsilon]^+ \underline{p}^x, \tag{2.9}$$

where + is used to denote the pseudo-inverse. Of course, this only works when the dimensionality constraints hold. Practically, they never do. This can be verified by doing characteristic vector analysis on the 1995 surface reflectances collected by Barnard et al. (2002). The number of basis functions required to characterise these measurements is substantially more than 2, and closer to 6 or 7. Consequently, this approach to colour constancy (although mathematically elegant) has little practical application.

2.4 Machine Learning

Machine learning techniques have been used to try and relate a scene to a given illuminant automatically. If we have a large set of scenes, and assume that every scene has a unique known illuminant then this can be used as the basis for training. Cardei et al. (2002) investigated using neural networks to solve for the colour of the scene illuminant. Binary colour histograms were made of each of the training images in chromaticity space. The quantity of pixels were ignored, so a chromaticity bin had a 1 if it was present in the scene and 0 if it wasn't. These binary relations were paired with the chromaticity of the scene illuminant. Xiong and Funt (2006) finessed this approach using support vector regression. Ning et al. (2009) also examined the use of support vector regression but on image derivatives.

Gijsenij and Gevers (2007) used the Weibull parameterization (e.g.texture and contrast) to separate images in a dataset into different classes. They found that different classes paired best with certain types of illuminant estimation algorithms.

2.4.1 Bayesian Colour Constancy

Several approaches are based on Bayes' theorem (Brainard and Freeman, 1997; Finlayson et al., 2002b; Gehler et al., 2008; Rosenberg et al., 2003; Sapiro, 1999). In general terms, the aim is to estimate parameters described by the the data A. The probability density of A is $\Pr(A)$. The function $\Pr(B|A)$ is the probability density of data B given A. We can refer to $P(A)$ as the prior probability and $\Pr(B|A)$ as the likelihood. The posterior probability is calculated using Bayes' rule

$$\Pr(A|B) = \frac{\Pr(B|A)\Pr(A)}{\Pr(B)}. \tag{2.10}$$

Brainard and Freeman (1997) calculated the prior distribution as the probability that a particular light and surface exists in the world ($\Pr(A)$). The posterior distribution was calculated using the image data. This was then used to generate a maximum likelihood estimate for the illuminant. The nature of how this algorithm is formed, means that illuminant estimates can be affected by the collection of surfaces in a scene. The model used in this work used a truncated probabilistic model. Rosenberg et al. (2003), in later work, used a non-Gaussian model to more accurately model the probability distributions. The approach combines the principle of the Bayesian framework with non-parametric image models, which assume a correlation of colour among nearby pixels.

2.5 Gamut Mapping

One of the most elegant colour constancy algorithms was developed by Forsyth (1990). He observed that the colours we can see in the world are limited by illumination. This means, for example, that under daylight, there are certain colours that a camera sensor would be very unlikely to capture.

He argued that the set of plausible colours for a given illuminant is constrained to a convex set in RGB space. Convexity holds as, given any two colours in the set, it is possible to make any colour that lies on the line between them. This is because the spectra of the reflectances merge in an additive way when they are captured by a sensor. This effect is demonstrated by a Bayer Mosaic in Figure 2.1.

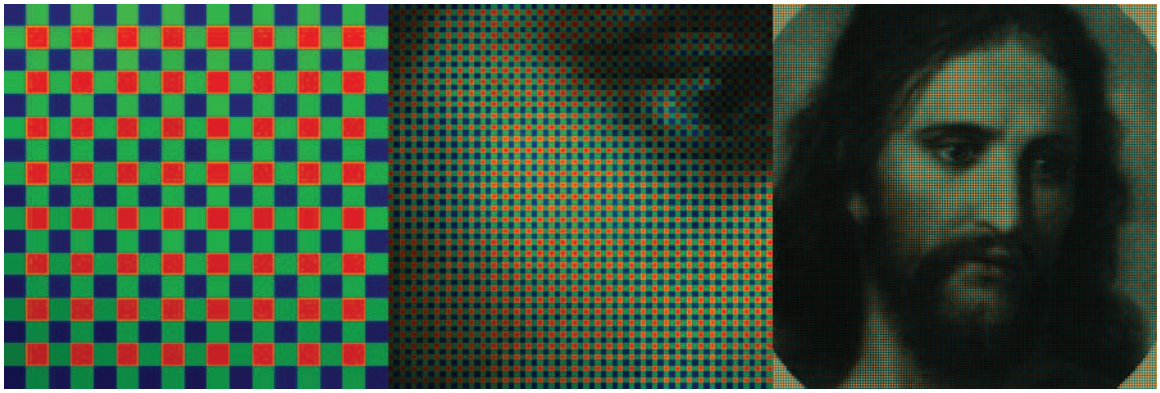

Figure 2.1: A Bayer Mosaic demonstrates the effect of the convex combination of three colours (rgb). We perceive a wider gamut of colours than just the different intensities of Red, Green and Blue.

Forsyth's algorithm assumes the diagonal model of image formation (Equation 1.2). If the illuminant of a pixel RGB was a uniform white light then the illuminant matrix $\boldsymbol{E}$ would be the identity matrix. A pixel under the canonical uniform white illuminant is equal to $\underline{S}_x$. We write the illuminant mapping matrix from a given illuminant to a canonical illuminant as $\boldsymbol{E}^{-1}$.

Now, suppose we have a large representative set of surface RGBs under the canonical illuminant then we can represent the set by it's convex closure. This set is characterised by the points on the convex hull. We denote the i^{th} point on the convex hull as $\Gamma(\mathcal{P})_i$. The mapping which maps an image RGB $\underline{p}$ to the i^{th} point on the canonical set is equal to

$$\mathcal{D}_i = \frac{\Gamma(\mathcal{P})_i}{\underline{p}}. \tag{2.11}$$

The set of all maps that take a single RGB into the canonical set, is also a convex set. We can define the scene RGBs as $\mathcal{I}$ and the j^{th} point on their convex hull as $\Gamma(\mathcal{I})_j$. For each vertex in the set, there is a convex set of mappings that will map vertices into the canonical set $\Gamma(\mathcal{P})_i$. We can think of this as a set containing all of the convex mapping sets. This set of sets $\mathcal{E}_j$ is described as

$$\mathcal{E}_j = \{\mathcal{D}_i : \mathcal{D}_i\Gamma(\mathcal{I})_j \in \Gamma(\mathcal{P}), \mathcal{D}_i = \frac{\Gamma(\mathcal{P})_i}{\Gamma(\mathcal{I})_j}\}. \tag{2.12}$$

The intersection of all of the sets in $\mathcal{E}$ is the set of mappings which will map $\mathcal{I}$ entirely into $\mathcal{P}$. The final set of mappings $\mathcal{M}$ is defined as

$$\mathcal{M} = \bigcap_{j=1}^{n} \mathcal{E}_j, \tag{2.13}$$

where n is the number of vertices required to define the convex set $\mathcal{I}$, which is subsequently the number of mapping sets in $\mathcal{E}$. The final step of the algorithm is to choose a single answer from the set of mappings. Forsyth proposed choosing the mapping that would maximise the volume of the mapped set (i.e. the mapping which made the most colourful image).

2.5.1 Gamut Mapping with Chromaticities

Finlayson (1996) argued that recovering the magnitude of the RGB of the illuminant

was implausible and unnecessary. For example, it maybe difficult to determine between a bright surface and a dark illuminant or a dark surface and a bright illuminant, therefore he argued that we don't expect to recover the illuminant intensity. With this in mind, it seemed natural to work in a chromaticity space. He proposed using the spectral-band ratio chromaticity space $[p_R/p_B \quad p_G/p_B \quad 1]$ (Equation 1.8). A colour constancy algorithm using this space will recover only the illuminant chromaticity. The steps of the algorithm can be applied straight from the RGB implementation to 2D chromatcity space. This is because a diagonal mapping in 3D RGB space has a unique 2D mapping in chromaticity space.

2.5.2 Gamut Mapping in a Bayesian Framework

The Gamut Mapping algorithms discussed so far output a set of illuminant mappings where the likelihood of each map being the correct answer is unknown. Finlayson et al. (2002b) proposed formulating this algorithm differently. They set out to recover a measure of the likelihood that each of a set of possible illuminants was the scene illuminant. Since they do not expect to recover the intensity of the illuminant, processing is done in a chromaticity space. For this application, the authors propose the rgb chromaticity space from Equation 1.6. This is because the space is fairly specially uniform.

For a given image under some known light we can build a 2-dimensional histogram (with N bins) which counts each occurrence of a chromaticity. Most sensor systems are 8-bit so have a range of 0-255. Given that chromaticities are 2-dimensional then we have a maximum of $N = 255 \times 255 = 65025$ bins to represent all the possible chromaticities. Although bins can be made larger. For a given image we now have an N element vector $\underline{v}^E$, representing the counts of chromaticities under a given illuminant E. We would like to know the probability of observing the i^{th} chromaticity

$\underline{c}_i$ under illuminant E. By using the data in $\underline{v}^E$ we can compute

$$\text{count}(\underline{v}^E) = \sum_{i=0}^{N} v_i^E, \tag{2.14}$$

$$\Pr(\underline{c}_i|E) = \frac{v_i^E}{\text{count}(\underline{v}^E)}. \tag{2.15}$$

Suppose we now take an image of an unknown scene and capture a single surface chromaticity $\underline{c}$. We would like to know the probability of E being the illuminant given that we recorded $\underline{c}$. We compute this using Bayes rule

$$\Pr(E|\underline{c}) = \frac{\Pr(\underline{c}|E)\Pr(E)}{\Pr(\underline{c})}. \tag{2.16}$$

If we assume that the points of all image chromaticities are independent and that all illuminants are equally likely then for an unknown input image C_{im}, we can write the probability as

$$\Pr(E|C_{im}) = \alpha \prod_{\forall \underline{c} \in C_{im}} \Pr(\underline{c}|E), \tag{2.17}$$

where α is some constant. We finally define a likelihood function

$$\text{l}(E|C_{im}) = \sum_{\forall \underline{c} \in C_{im}} \log\left(\Pr(\underline{c}|E)\right). \tag{2.18}$$

The log probabilities measure the correlation between chromaticites and an illuminant. Finlayson et al. (2002b) proposed forming this in a correlation matrix where ij^{th} elements are $\log\left(\Pr(\underline{c}_i|E_j)\right)$. By summing the rows of this matrix we get a likelihood for each illuminant. We can then choose the illuminant with the maximum likelihood as scene illuminant estimate. This is shown in Figure 2.2.

This algorithm is computationally simpler than a full implementation of Gamut Mapping, and includes more probabilistic information about the illuminants. However, the calibration step requires a large amount of input data.

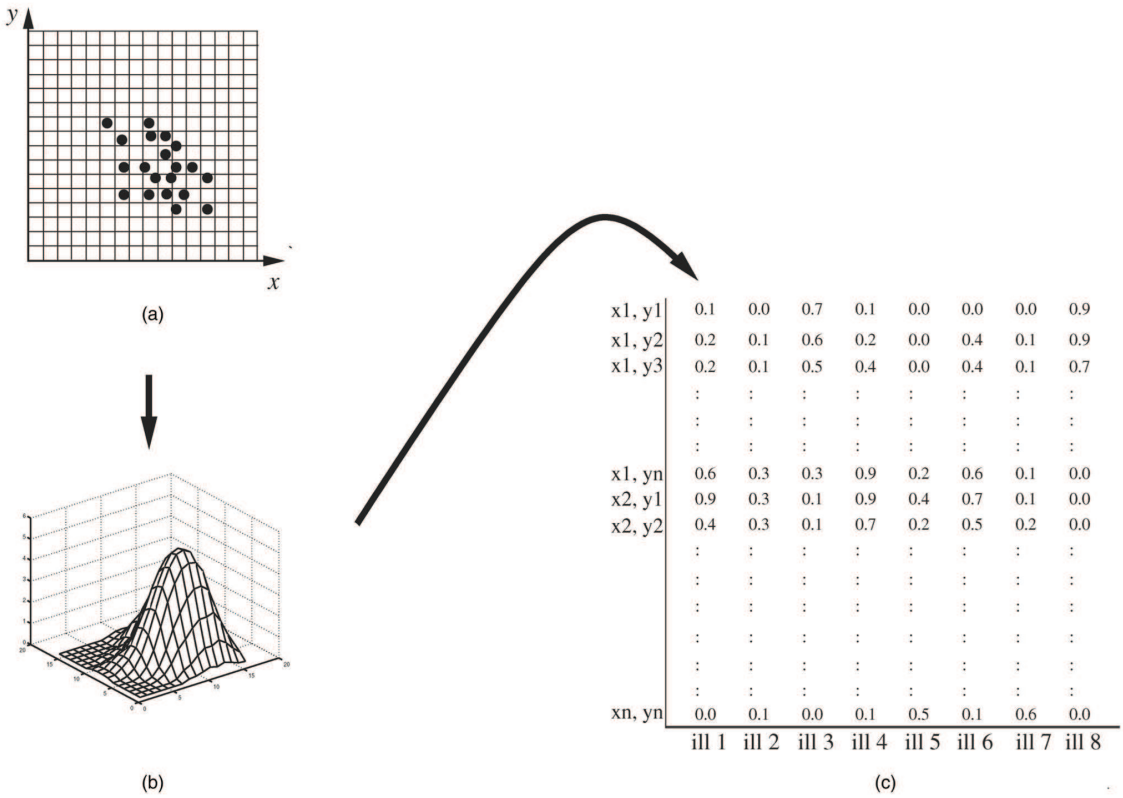

Figure 2.2: Three steps in building a correlation matrix. (a) We first characterise which image colours (chromaticities) are possible under each of our reference illuminants. (b) We use this information to build a probability distribution for each light. (c) Finally, we encode these distributions in the columns of our matrix.

2.6 Physics-Based Algorithms

The algorithms presented in the previous section use the Lambertian model of image formation (shown in equation 1.1). This model is only a partial representation of how a pixel could be formed. Several authors (Lee, 1986; Shafer, 1985; Tan et al., 2004; Tominaga, 1996; Tominaga and Wandell, 1989) have proposed a dichromatic model of image formation. This includes an additional component to account for specularity. Surfaces are modelled as having a diffuse component, which reflects light evenly in every direction, and a specular component which is direct mirror-like reflection. A

perfect specularity is where all of the light is reflected and is unmodified by the surface. A pixel RGB at this point would therefore be representative of the illuminant. The dichromatic model of illumination is defined as

$$p_k^x = \int_\omega \left[m_d^x S^x(\lambda) E(\lambda) R_k(\lambda) + m_s^x E(\lambda) R_k(\lambda) \right] d\lambda, \tag{2.19}$$

where m_d^x and m_s^x are the relative weights associated with surface x. It is straightforward to show that when the diffuse weight m_d^x is zero then p_k^x represents the illuminant RGB. It follows that if we can find specular highlights in images then we can use them to estimate the illuminant. A uniform surface with a specularity will have RGBs that lie on a plane according to 2.19.

The pixel RGBs are combinations of diffuse and specular component RGB vectors. By decoupling this information, the illuminant can be derived. Shafer (1985) observed that specular information on more than one surface will generate two 2D planes. It follows then that the intersection of these planes will be the component which they have in common, which is the specular illuminant component. Unfortunately, this kind of algorithm only works in laboratory conditions, and performance isn't as good when used on real images. It is also not easy to segment two surfaces that have specular variation.

In other work Funt et al. (1991) tried to use the mutual inter-reflection between two surfaces to provide more information about the illuminant colour. Mutual reflection occurs when light reflected from one surface is incident upon a second surface. They identified mutual reflection regions and analysed how the sensor responses changed with the aim of recovering only the surface reflectance RGB. This could then be used to recover the illuminant. This algorithm, like others in this category is hard to apply in real-world scenarios and only has good performance in laboratory conditions.

2.7 Specialist Cameras

Finlayson et al. (2005a) proposed a camera which took two images of every scene, which they called a Chromagenic Camera. One image would be a regular camera image, the second would be the same image taken through a colour filter. The theory behind this work is unusual, relative to other work because it relies on the sensors not being completely narrowband. The narrowband assumption, which leads to the diagonal model, is an approximation for image formation in most cameras. If sensors were completely narrowband then the effect of a filter could be modelled by a diagonal transform. Assuming the sensors are not completely narrowband, then the mapping from a filtered image to an unfiltered image would require more degrees of freedom than a diagonal mapping would give. The authors show that the 3×3 mapping matrix RGBs in a filtered image to an unfiltered image changes as the illuminant does. In fact, when surface and illuminant spectra are 3-dimensional then the mapping matrix can uniquely identify the illuminant. However, as explained in section 2.3, surface reflectance requires more degrees of freedom than 3 to be modelled accurately. Even when this condition does not hold the algorithm has shown good performance on real images, but does have significant failure cases.

In an extension of this work Finlayson et al. (2005b) looked at the performance of the chromagenic algorithm using different filters. In their work, they proposed a method for filter design, which was optimised specifically for a Colour Constancy application.

Skaff et al. (2009) formulated the the filter/non-filter approach in a more generalised probabilistic framework. Rather than requiring every pixel to be a pair of filtered/non-filtered pixels, they allowed for this information to be incorporated into the Bayesian probabilistic model. The extra information reduces the effect of the Prior in Bayesian Colour Constancy and produced more accurate results.

Fredembach and Finlayson (2008) later looked at improving the chromagenic algorithm. They noticed that brighter pixels in the image produced better mapping matrices than using all of the pixels. The modified algorithm simply involved taking the top 10% brightest pixels instead. The improved performance may simply be due to brighter pixels being less noisy.

Finlayson et al. (2007) found that the chromagenic filter could be applied to finding parts of the images lit by different lights.

2.8 Varying Illumination

This school of algorithms aims to simultaneously solve for multiple illuminants. This has applications in scenes with shadows, because a shadow often means a change in illumination colour. Barnard and Finlayson (1996) aimed to identify and remove illumination variation in images. The authors proposed a segmentation approach based on Retinex (Land and Center, 1974). They used illumination change directly to improve on Gamut Mapping (Forsyth, 1990), which in it's original form assumes input of surfaces under a single illuminant. In this section we look at additional work that uses varying illumination information in order to estimate illumination colours.

2.8.1 Finite Dimensional Linear Models

DZmura and Iverson (1993a,b, 1994) in their three-part paper on Colour Constancy extended the approach of Maloney and Wandell (1986). The authors examined conditions in which spectral information could be inferred from surface reflectances and illuminants. They showed that spectral recovery was possible in the following conditions: number of sensors $\geq$ illuminant model dimensions $\geq$ number of illuminants and the numbers of surfaces = the surface reflectance model dimensions. This means that for a tristimulus system the same surface under 3 illuminants would be required.

Finding the same surface in a scene under 3 illuminants is rare, also more than three basis functions are required to properly characterise illumination change (Barnard et al., 2002). This model is theoretically interesting but has little practical application.

2.8.2 We need to talk about Kelvin

We know that visible light is a form of radiation in which electric and magnetic energy oscillate at wavelengths roughly between 400 and 700 nanometers. We can increase the amount of radiation emitted by an object by increasing it's energy (heating it up). At different temperatures we will perceive different colours, from the reddish flame of a wood burner, to the yellow/white of the much hotter sun. So there is an intrinsic link between energy and spectral radiance.

In physics, it is classic to study this behaviour in perfect blackbodies. Where a perfect blackbody is a theoretical object that absorbs all radiation that it comes in to contact with (it reflects nothing). All objects in the universe emit some blackbody radiation, however they have to be very hot for them to emit in wavelengths that are detectable by our eyes. At 3000°Kelvin(K) the peak radiance is outside of the visible spectrum, it lies in the infrared. The tail of the radiance that we can perceive will appear red. As it cools it will not appear to radiate at all.

In the early 20th Century Max Planck was able to describe the radiation of blackbodies. He found that as the temperature increased, the spectrum's peak moves towards the bluer part of the spectrum. This is demonstrated in Figure 2.3.
Most lights are orange, yellow, white or blue. These are the same colours that occur when metals are heated to different temperatures. In Physics, this was encapsulated as Planck's law, written

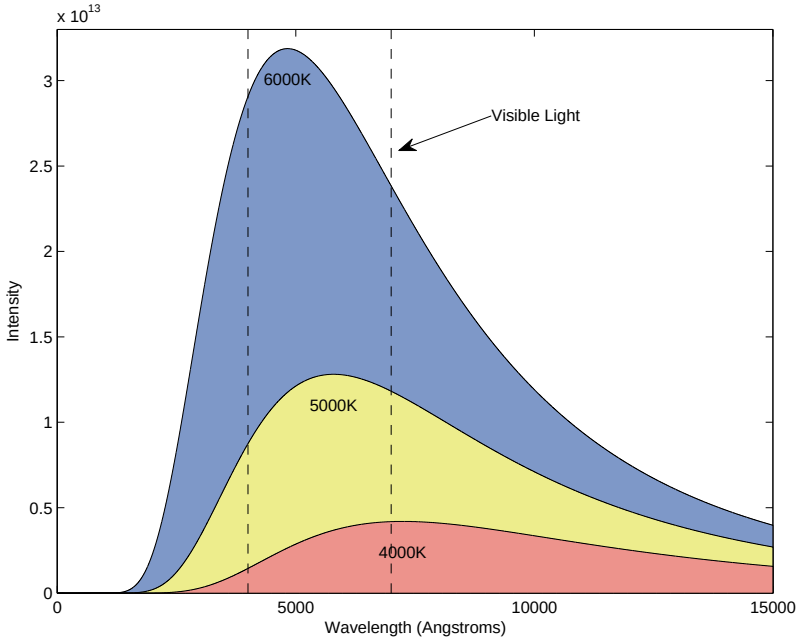

Figure 2.3: Graph shows the spectra calculated from Planck's equation of blackbody radiation at different temperatures. The peak is shifted to the bluer part of the spectrum as temperature increases.

$$E_T(\lambda) = \frac{2hc^2}{\lambda^5} \frac{1}{e^{\frac{hc}{\lambda kT}} - 1}, \tag{2.20}$$

where T is temperature, k is the Boltzmann constant which relates energy at the individual particle level to temperature, h is the Planck constant which relates the energy of a photon to the frequency of its associated electromagnetic wave and c is the speed of light. In the world, our main source of natural illumination is the sun. The sun is an almost perfect blackbody radiator. This is shown in Figure 2.4, where the sun's spectral emittance has been plotted against a closely matching blackbody emittance according to Planck's law.

In an attempt to understand the colour properties of blackbody radiators, we

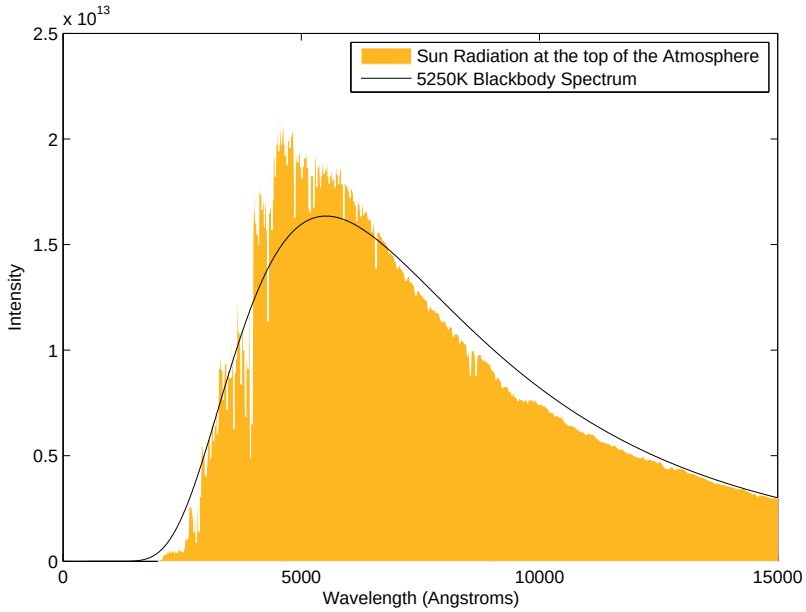

Figure 2.4: The sun is an almost perfect blackbody, plotted here with planck's formula evaluated at 5250°K.

can integrate them using tristimulus sensitivity functions. The CIE colour matching functions were derived to match colours from different colour spaces to human perception. The colour space produced by these sensitives is called XYZ. We can normalise the channels by their sum to describe colour independently of brightness. Chromaticities generated from Planckian illuminants are often plotted as a locus on the xy chromaticity space diagram (as shown in Figure 2.5).

Several works (Henderson and Hodgkiss, 2002; Judd et al., 1964; Kawakami and Ikeuchi, 2009; Wyszecki and Stiles, 1982) investigated the spectral properties of daylight have reported how remarkably close daylight chromaticities are to the Planckian locus. A daylight locus, generated from real measured daylights was plotted by Wyszecki and Stiles (1982). It is almost exactly parallel to the Planckian locus. The varying illuminants generated by this process are represented in the photograph in

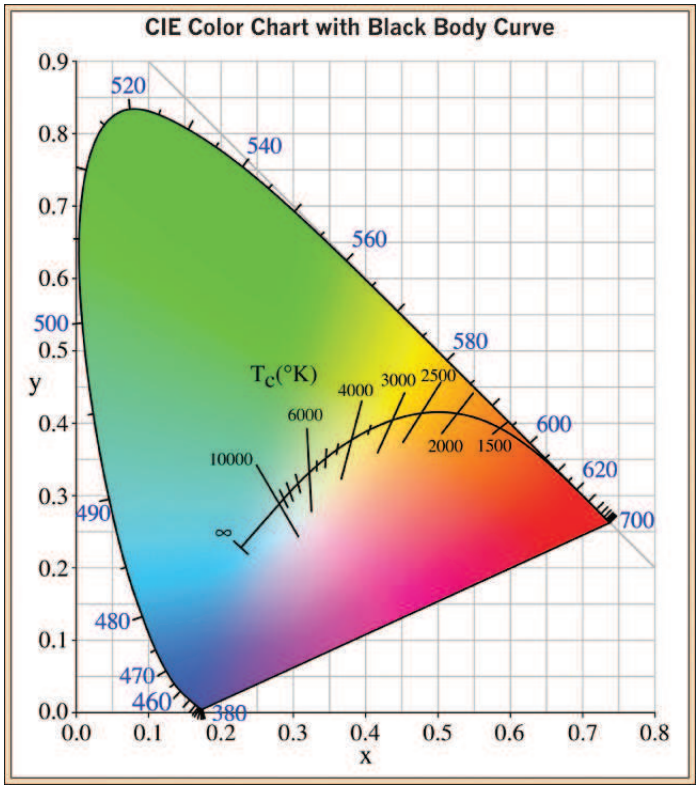

Figure 2.5: The CIE XYZ chromaticity diagram with Planckian Locus. Where x = X/(X+Y+Z) and y = Y/(X+Y+Z).

Figure 2.6.

White balancing tools in software (such as that found in Apple's Aperture and iPhoto packages) use colour temperature as a guide for the user to find their preferred white point. The locus forms a highly constrained set of illuminant colours. If a Colour Constancy algorithm produced a green illuminant (as GreyWorld might when calculated on a very leafy scene), we would be able to reasonably access it's accuracy by measuring how close it is to the locus.

2.8.3 The Illumination Mapping Line

Most colour constancy algorithms use the varying surfaces in a scene to try and make assertions about the scene illuminant. Here we are interested in the varying

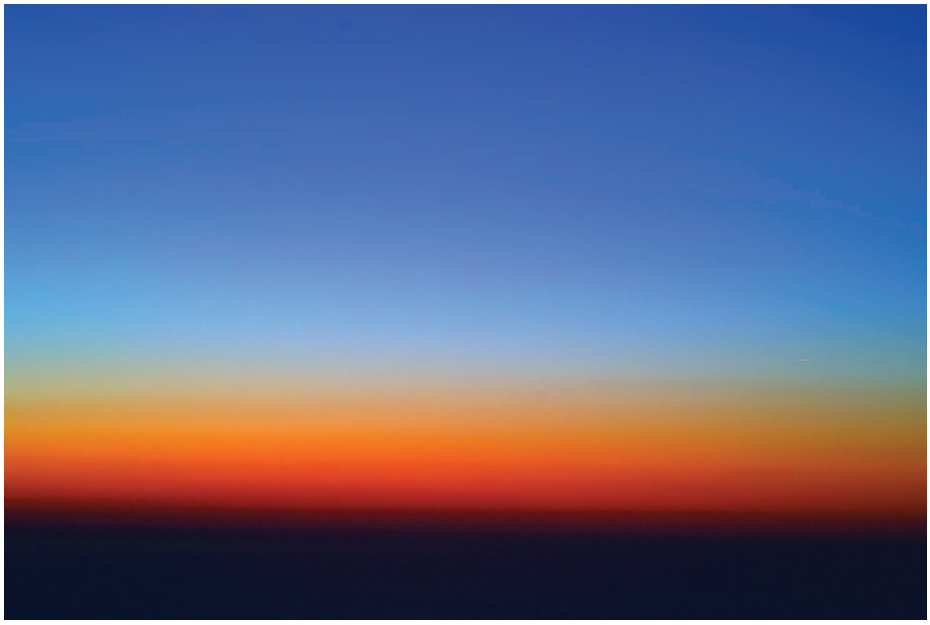

Figure 2.6: Photograph taken 1 hour after sunset at 500 meters altitude.

illuminants in a scene. Planck's law is often used as a model for real-world illuminant colours. To demonstrate the validity of this, we plot a set of real world illuminants measured in Granada, Spain (Hernández-Andrés et al., 2001) in chromaticity space. This is plotted along with the Planckian Locus. This is shown in Figure 2.7. The real world illuminants fall remarkably close to the locus.

Finlayson et al. (1995) looked at using this apparent 1-dimensionality of illuminant colours in a colour constancy algorithm. From Equation 1.5, we can write the diagonal illuminant matrix $\boldsymbol{E}$ as an illuminant mapping $\boldsymbol{E}^{-1}$. This also holds in spectral-band chromaticity space. The notation below assumes this chromaticity space because we do not expect to recover illuminant brightness.

The mapping matrix $\boldsymbol{E}^{-1}$ takes pixel chromaticities to surface reflectance chromaticities. We can observe that Planckian illuminants roughly lie on a line when plotted in this space. We have shown this previously in Figure 1.7, and the theory is

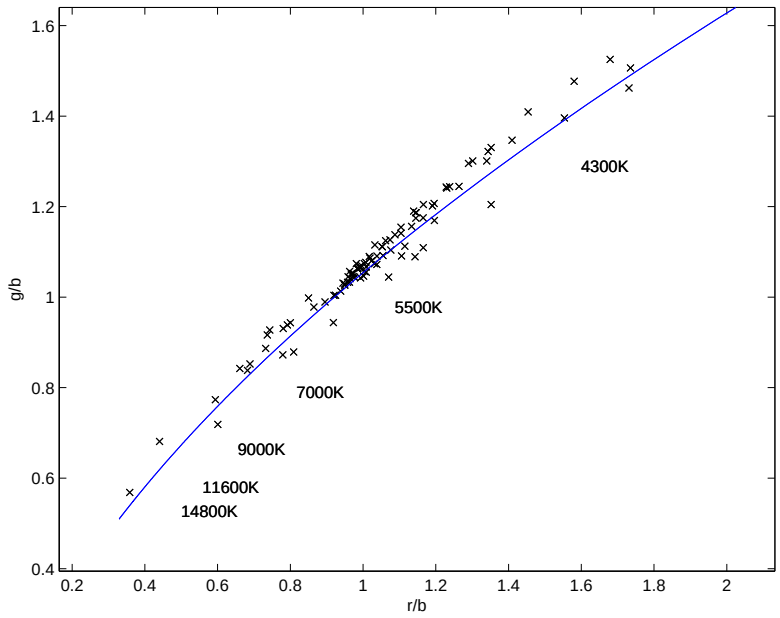

Figure 2.7: Illuminant Spectra captured by Hernández-Andrés et al. (2001), plotted in chromaticity space along with the Planckian Locus.

further supported when plotting the Planckian Locus in this space (Figure 2.8).

Real-world illuminants roughly occur between the colour temperatures of 3500K to 12000K. We can see that the locus is well-approximated by a straight line here. That the mapping to the reference illuminant can be represented well by a line is used by Finlayson et al. (1995) to solve for the colour of a surface at an illuminant edge. They take the line of best fit to represent the set of all possible mappings. Let us use Φ to denote this set. The set $\underline{p}^E\Phi$ consequently describes all the possibilities for $\underline{S}$ (where the subscript E indexes the E^{th} illuminant). In an outdoors scene there is likely to be the same surface captured under two lights. We can find this at a shadow edge. We can represent these two pixel responses as $\underline{p}^1$ and $\underline{p}^2$. Given only these two responses Finlayson et al. (1995) showed that we can calculate $\underline{S}$ with the equation

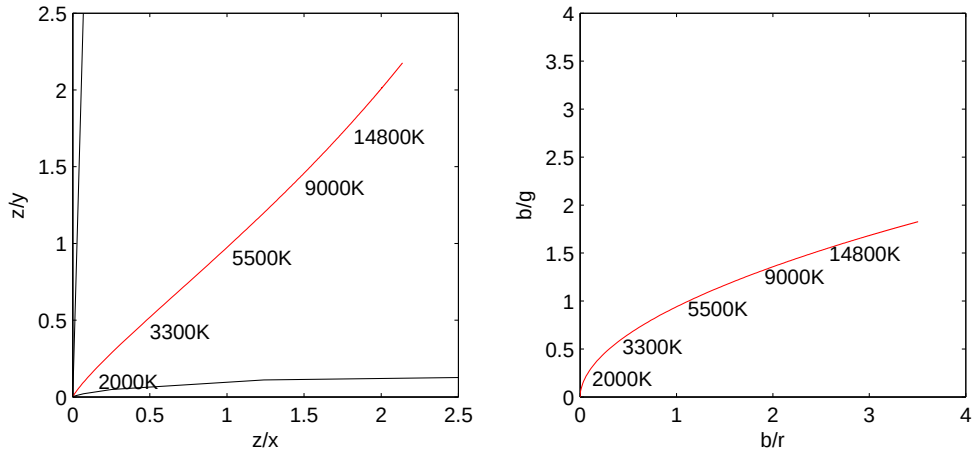

Figure 2.8: **Left:** Planckian locus in inverse chromaticity xy space **Right:** Planckian locus in inverse chromaticity Sony-DXC 930 camera space

$$\underline{S} = \Phi \underline{p}^1 \cap \Phi \underline{p}^2. \tag{2.21}$$

The intersection of the two mapping sets $\Phi \underline{p}^1$ and $\Phi \underline{p}^2$ will always yield a unique solution, as only one possibility for $\underline{S}$ will be common to both sets. This approach makes colour constancy possible with minor calibration and only 2 pixels for input.

2.8.4 The Illumination Mapping Line Segment

Preliminary work investigated the benefit of using varying illumination as a cue for Colour Constancy. We have learned that natural illumination behaves predictably and is constrained. In the previous section we explained how the work of Finlayson et al. (1995) attempted to use this quality. The resulting algorithm requires minor calibration, little input and is computationally simple. Unfortunately, it is prone to error. Kawakami et al. (2005) investigated this method, in their attempts to recover true surface information for a graphics application. They concluded that there were common scenarios which caused this method to fail. In fact, it is possible for the

algorithm to output a negative answer, which clearly is not a valid result. They argued that an infinite line isn't a valid representation of the illuminant set. The Planckian locus is constrained in chromaticity space, so it makes sense to constrain the illuminant set also. The authors highlighted that it was sensible to represent the set as a line segment instead. The only difference this change has, is that it is possible for the lines to not intersect. This happens where no two illuminant mappings both lie on the line segment, for the given pixel inputs. The authors suggest the following reasons for error:

1. The diagonal model does not hold.
2. The correct illuminant mapping does actually lie outside of the line segment.
3. The pixel data contains noise

Kawakami and Ikeuchi (2009) formulated a colour constancy algorithm that tried to address these issues. They assume that the first 2 circumstances are not the problem, and investigate a possible approach given noisy data. The authors start with the assertion that the correct illuminant will lie on the line segment. They calculated the end points by analysing the probabilities of a scene being lit by a given illuminant. They found that colour temperatures in the range of 3500K to 12000K represents 4 standard deviations of the variation. If we look back to Figure 2.8 we can see that this range falls on the most linear part of the the locus. The authors introduced the constraint that any light mapping output from the algorithm should lie on the line segment. To force an intersection and maintain this condition they proposed adjusting one of the pixel responses. This becomes a translation of one of the line segments. If we assume that the location of the first illuminant lies at the ratio $u : u - 1$ on the illuminant mapping line, then it doesn't matter which pixel chromaticity is adjusted as long as the ratio $u : u - 1$ is maintained. Constraining

this property is what makes the adjustment a translation. This is shown in Figure 2.9.

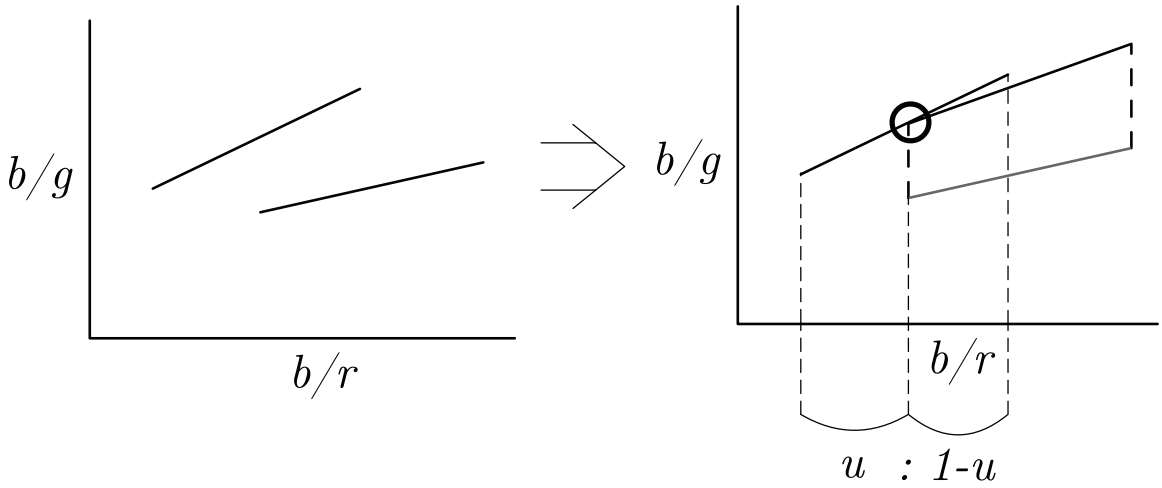

Figure 2.9: Kawakami and Ikeuchi (2009) method of line segment intersection. The ratio $u : u - 1$ determines the 2 illuminants. It is maintained no matter which line segment is adjusted.

This algorithm does have edge-cases where it cannot output a solution. This is when there is no intersection even after a translation in one of the axis. If the line segment covers a sufficient space of illuminant colours then this is less likely to happen. In Chapter 4 we show how we can formulate the intersection of line segments in a mathematically optimal way, and consequently get much better colour constancy performance.

2.8.5 Using Traditional Algorithms

Colour constancy algorithms have traditionally been applied to a scene with the assumption that it contains surfaces that are only lit by a single illuminant. This is obviously not always the case. M Bleier et al. (2011) created a dataset of images containing objects which were lit by 2 to 3 illuminants. After capturing these images, the objects were spray-painted with a matte white coat to create a ground truth

image of the illumination. The authors split images into "superpixels" of similar colour. They ran traditional colour constancy algorithms on each "superpixel" and compared the answer to ground truth. They found algorithm's do not perform as well in cases of multiple illuminants.

In later work Gijsenij et al. (2012) proposed using different methods of image segmentation. These included grid-based, key-point sampling (Mikolajczyk and Schmid, 2004) and segmentation using edges (Comaniciu and Meer, 2002). The authors suggest that modelling each segment as a surface under a single uniform illuminant is much more likely to be accurate than applying this assumption to the entire image. For edge-based segmentation this will particularly be true because illumination changes caused by shadows will be segmented out. However, this approach means that there is a lot less surface variation. The algorithm involves running a traditional colour constancy algorithm on each patch (such as GreyWorld (Buchsbaum, 1980), GreyEdge (Weijer and Gevers, 2007) or Gamut Mapping (Forsyth, 1990)). An illuminant estimate for each patch is output. The authors make the assumption that all the illuminant estimates for patches under the same illuminant will cluster together. If the number of illuminants in a scene is known, then using a clustering algorithm (such as k-means (Alsabti et al., 1997)) the patches can be grouped together. The patches considered to be under the same illuminant can then be combined and put through the algorithm again to get a more robust estimate for each illuminant.

Performance for grid-based segmentation was shown to be the best. Grid-patches will contain more surface variation than edge-based segmentation as boundaries are arbitrary placed. Algorithms such as GreyWorld and MaxRGB perform very badly when there is low surface variation.

The work of this thesis focuses on the hard cases of images with varying illumination. The change in illumination is used directly in order to derive information about

the illumination colour. In the next few Chapters we propose several new Illuminant Estimation algorithms which focus directly on the case of multiple illuminants. We discuss their merits and disadvantages and evaluate them experimentally against other work. We have found using information from more than one illuminant is very powerful for illuminant estimation, and performance is shown to be better than our predecessors.

Chapter 3

Datasets for Illuminant Estimation

3.1 Overview

Illuminant Estimation algorithms are traditionally approached as a system which takes an image as input and outputs a single RGB colour that is the estimate of the illuminant. Strictly, this system only has valid application where a single uniform illuminant is applied to the scene. In a lot of scenarios this is not the case. We examine one of the most popular datasets for Colour Constancy research, provided by Gehler et al. (2008). This dataset contains 568 images of different indoor and outdoor scenes. Every image contains a colour chart, with 6 grey-patches (used to infer the correct illuminant). This has become the dataset of choice for benchmarking modern algorithms (Bianco and Schettini, 2012; Finlayson, 2013; Gijsenij et al., 2011).

In this chapter we provide a new linear rendering of the Gehler et al. (2008) dataset. This dataset has previously been processed to be linear by Shi and Funt (2009), however there is some confusion in the rendering (the authors have updated their website on at least 2 occasions and now the community is using different versions of their dataset).

We also provide two new datasets which are designed specifically to be hard for current illuminant estimation algorithms. Both datasets contain more than one illuminant colour in every image (i.e. contain noticeable shadows). The first dataset has

a very small amount of surface variation (only 5 surfaces in each image). The second set of images are closer to the images people would capture in the real world. They contain different scenes with predominant shadows in all of the images.

3.2 Introduction

To be able to assess the performance of a Colour Constancy algorithm, the variables it is trying to estimate must be known in advance. At a pixel level, we know that the RGB response is produced out of an intimate relationship between the colour of the illuminant and the reflectance properties of the surface. By changing the lighting conditions in a room you are also changing the physical colours perceived by your vision system. Cancelling out this effect means that your vision system can make more accurate decisions based on surface colour. When photographs of outdoor scenes are used as the input to these algorithms, the assumption is that there is a single correct white point.

From a photographers perspective, performing a white balance correction involves scaling the RGB channels until the image no longer looks like it has a colour cast. It is common for software (such as Apple's Aperture and Adobe's Lightroom) to constrain the scaling such that image brightness is preserved. The goal of an automatic white balancing algorithm would be to estimate the scaling (the inverse of the white point) that leads to the most pleasing photo. This goal is often confused with recovering the illumination colour in the scene. In fact, posed as just described, recovering the illumination colour in a scene can sometimes be an impossible task. This is because not all scenes are lit with a single uniform illuminant. This is particularly true outdoors, where objects in shadow are physically bluer than objects out of shadow. This concept has been understood by artists for centuries, who will paint objects in shadow to not only be darker, but also bluer. An image appears correct to us

if a white surface appears white in sunlight, but the same white surface is bluer in shadow.

Figure 3.1: The Blue Shadow by Kathleen Harris painted in 1963

3.3 Current Datasets

In this section some of the many colour constancy datasets are considered. We will discuss their merits and disadvantages with regards to multiple lights and shadows.

3.3.1 Synthetic Data

One of the ways to capture data that exactly conforms to the image formation model is generate it synthetically. We can approximate the image formation model, by

measuring spectra at discrete intervals, and substituting the integral (from Equation 1.1) with a summation

$$p_k^x = \sum_{\lambda=1}^{n} S^x(\lambda)E(\lambda)R_k(\lambda). \tag{3.1}$$

Barnard et al. (2002) provided a large set of measured spectral data. Including a set of 1995 surface reflectances and the RGB sensitivities of the Sony-DXC 930 camera. These samples are measured in intervals as fine as 5 nanometers. These represent the functions $S^x(\lambda)$ and $R_k(\lambda)$ in the image formation equation. Hernández-Andrés et al. (2001) have provided a broad range of daylight spectra at different times of day and weather conditions in Granada, Spain. These can be used to make the $E(\lambda)$ function.

With this data, synthetic images can be formed where we know the assumptions of our model hold, with the added benefit that the underlying spectra have been recorded from real data. We also know the exact illuminant of every pixel, without having to account for the effects of a change in illumination (Gijsenij et al., 2012), specularities (Lee, 1986) or inter reflection (Funt et al., 1991). Although performance on synthetic data does not necessarily correlate perfectly with performance on real images, it is a reasonable place to start because algorithms that do not perform well using this data are very unlikely to perform better on real images.

For algorithms that require higher level data than just 'bags of pixels' some authors test using the hyperspectral images provided by Foster et al. (2006). In this work, the authors captured the full spectra of every pixel in an image. They also measured the spectra of the scene illuminant. This was divided out to get an image that roughly corresponds to only surface reflectances. This means that synthetic images can be generated that actually contain the content of real photographs. Then using Equation 3.1 we can render the scene under different lights, and test illuminant estimation

algorithms. We show an example rendering of a single hyperspectral image under 15 different daylights in Figure 3.2. Because they are daylights, the weather conditions and time of day look different in each image. The redder images look like sunset and the bluer images look like overcast conditions.

These images are generated synthetically but contain higher-level data. This means that algorithms that use derivative information (Gijsenij et al., 2009, 2010; Ning et al., 2009; Weijer and Gevers, 2007) could be tested on this dataset.

Figure 3.2: A hyperspectral image from the Foster et al. (2006) dataset rendered under 15 lights from the spectra measured by Hernández-Andrés et al. (2001).

3.3.2 A Large Dataset for Colour Constancy

Ciurea and Funt (2003) composed a dataset of 11,000 images, with the motivation that most colour constancy datasets aren't large enough. Attached to the camera was a uniformly grey ball, which is visible in all of the images. Taking the average of the grey-ball pixels provides an estimate for the white point. The images were captured with a Sony VX-2000 digital video camera. The images in this dataset are processed JPGs. This means they are not linear and contain compression artefacts. A selection of images from the dataset are shown in Figure 3.3.

To try and minimise the dataset containing many copies of similar images, only 3 frames per second were included from the 2 hours of footage. A selection of the images from this dataset are shown in Figure 3.3. In the paper the authors acknowledge that there is often more than one dominant illuminant colour on the ball. To demonstrate this we analyse the change in chromaticity over the ball in an image in the set. We project sum-normalised chromaticities (Equation 1.6) onto the plane perpendicular to [1 1 1] (to encourage spatial uniformity) to get

$$r = p_r - p_g, \quad g = \frac{p_r + p_g}{2p_b}, \tag{3.2}$$

where $r = 0$, $g = 0$ represent shades of grey (from black to white). In Figure 3.4 we model (in rg space) the distribution of chromaticities over the grey-ball. It is clear that the distribution contains two distinct peaks in chromaticity. For this image, this is likely due to the effect of indoor illumination from the ceiling lights, and outdoor ambient illumination from the windows. It is not obvious how a single white point should be selected from the ball given this distribution. However, provided with the dataset is a single illuminant RGB for each image. The value with the largest peak in the distribution on the ball is taken to be the illuminant colour (does not necessarily correlate with the dominant scene illuminant).

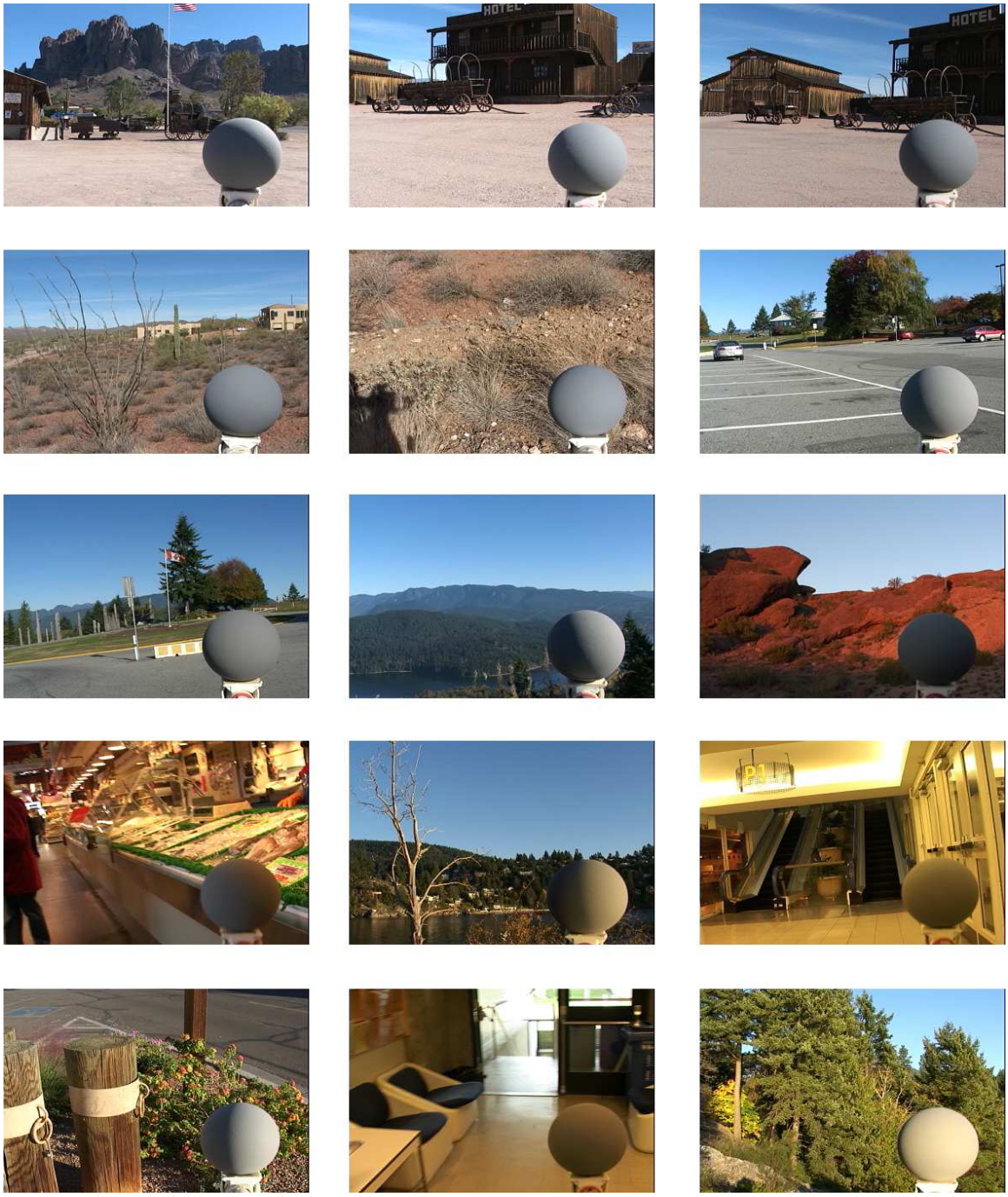

Figure 3.3: A selection of images from the Ciurea and Funt (2003) dataset.

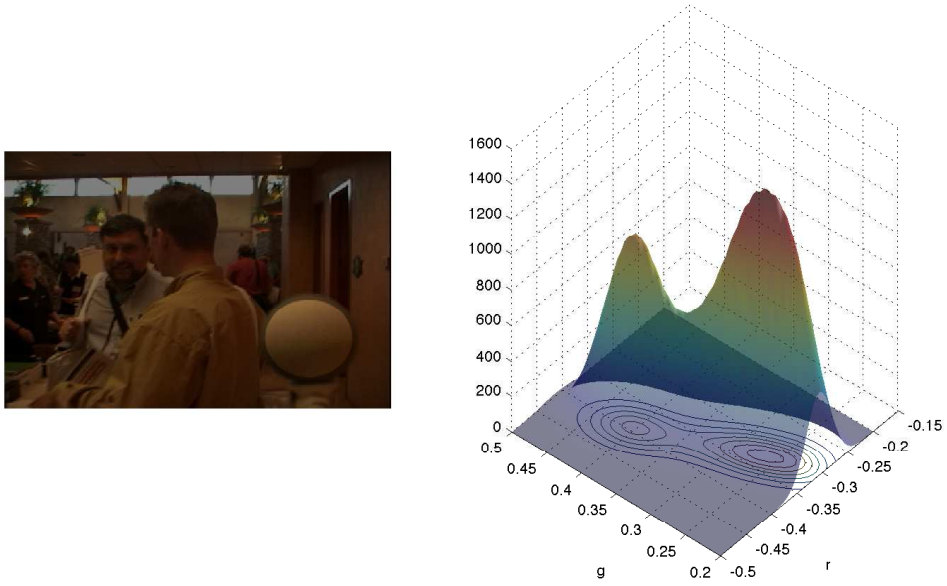

Figure 3.4: Distribution of Chromaticities from the pixels in the highlighted circular region on the grey ball. Shows two distinct peaks.

Figure 3.5 shows the distribution of lights captured in this dataset according to the light RGBs provided. In chromaticity space the light colours are roughly 1-dimensional. We regress these points onto a quadratic, so we can visualise the distribution easily. There is a large peak in the blue part of the spectrum, which means that a lot of the images were captured in similar lighting conditions, and probably at the same times of day. However, the breadth of images captured means that the distribution across the other light colours still accounts for a lot of images.

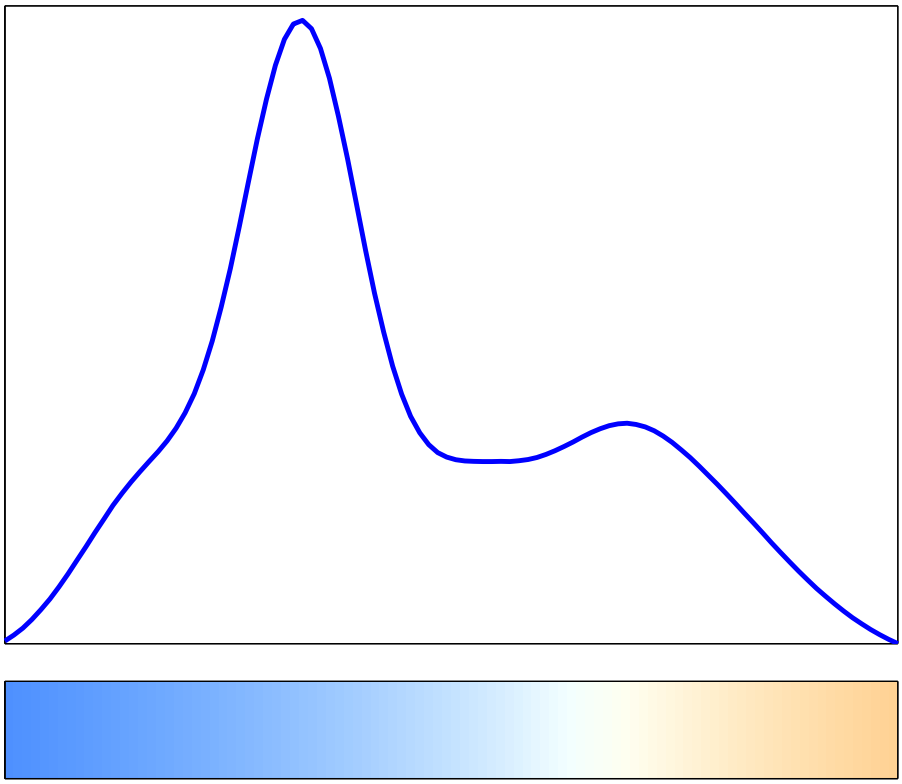

Figure 3.5: Distribution of the illuminant RGBs in the Ciurea and Funt (2003) dataset.

3.3.3 The Gehler et al. (2008) Dataset

Gehler et al. (2008) in their work, captured a dataset of 568 images taken at different indoor and outdoor locations around Cambridge, England. The dataset was captured using two Canon DLSR cameras (the Canon 5D and Canon 1D models). Every image contains a single Macbeth Colour Checker. The grey patches on the Colour Checker are used to provide the image white point. The authors provided the coordinates of

each patch on the colour chart in every image. This means that the pixels of the grey-patches can be easily extracted. In most images, the illumination is fairly constant across the scene. However, for some images there are stark shadows, and the Colour Checker is arbitrarily placed either in or outside of the shadow. There are often cases of indoor and outdoor illumination in the same scene.

The dataset was processed for display using the Canon software that came with the cameras. However, Shi and Funt (2009) have provided a reprocessing of the dataset, with the aim of maintaining the original counts of the RGB pixels from the camera sensor. However this processing of the images appeared desaturated and had a strong cyan colour cast (see Figure 3.6). Unfortunately, we found that images were not processed as originally stated. The black level had not been corrected for, which means the image pixels were a translation away from the correct values. We would like to note that the authors recently provided a Matlab script to correct this issue. [1]

To resolve the problem of the incorrect Shi and Funt (2009) rendering we provide a new processing of the images using the free open-source software DCRAW. The images were output in linear DOC-RAW TIFF format, which are greyscale images, that map directly to the response pattern of the Bayer camera sensor. These images were then demosaiced by resizing the image to 50% so the 2×2 pattern of RGBs was reduced to a single RGB pixel. The green value was taken as the average of the two green measurements. The black level in this version was corrected for. This removed the "washed-out" effect in the previous rendering [2]. The two different renderings are compared visually in Figure 3.6.

In Figure 3.7 we show the distribution of lights for this dataset (the same as in the previous section). The colours of the lights are slightly different here. This is due

[1] File http://www.cs.sfu.ca/~colour/data/process_568.m, create date: 9/16/2013 - modify date: 9/16/2013.

[2] Data at http://colour.cmp.uea.ac.uk/datasets/reprocessed-gehler.html

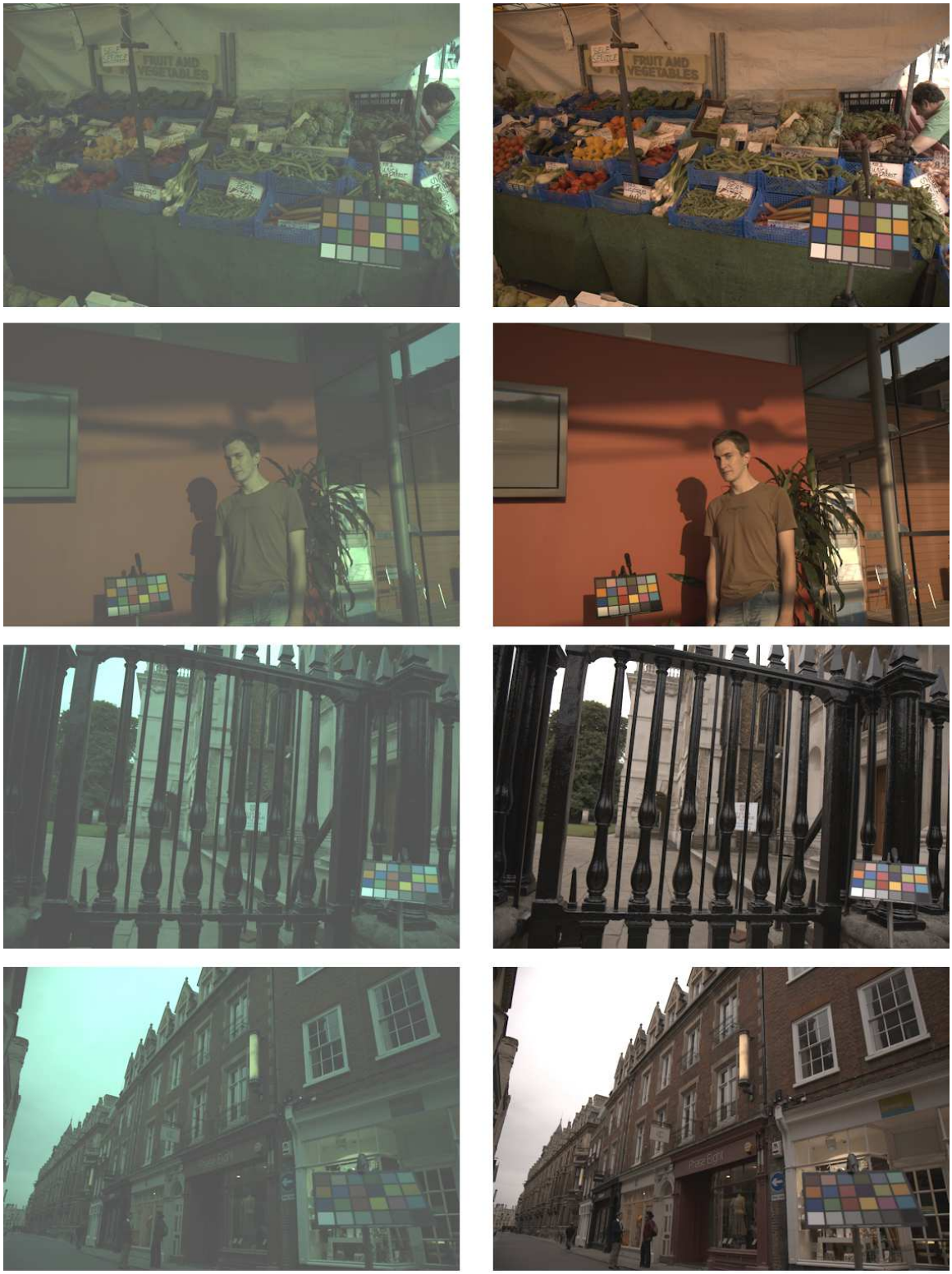

Figure 3.6: **Left:** Shi and Funt (2009) processing of Gehler et al. (2008) dataset. **Right:** Our Processing

to the different sensors and renderings of the two datasets. However, we see the same distribution of lights going from blues to yellows to whites to reds. The distribution has two distinct peaks, with the majority in the bluer part of the spectrum. The reddest illuminants are mostly captured from the indoor scenes in this dataset.

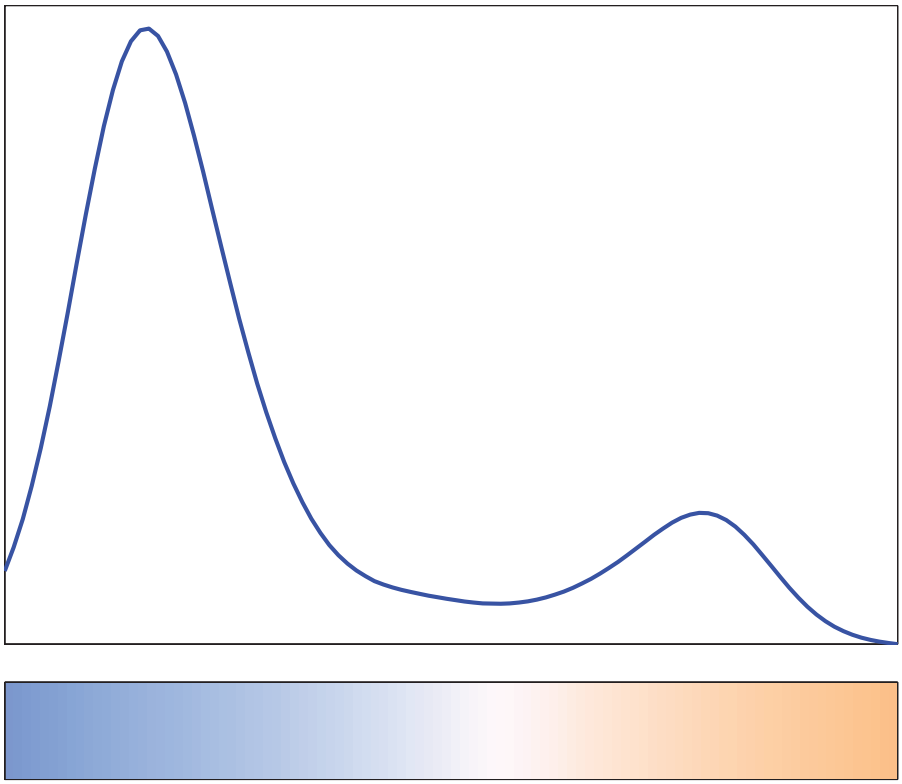

Figure 3.7: The distribution of lights in the Gehler dataset.

3.3.4 Multiple Illuminant Ground Truth Dataset

Recently M Bleier et al. (2011) were interested in how Colour Constancy algorithms performed under the presence of multiple illuminants. In this work, the authors acknowledge that there are many different scenarios where a photograph could be taken under more than one illuminant. They argue that image segmentation is a

required step to give these algorithms relevance. The ideal case would be to segment the image into areas of different illumination. The authors suggested segmenting an image into superpixels (Veksler et al., 2010). These are blocks in the image of roughly the same colour. The underlying assumption is that illumination is approximately locally constant in each superpixel. An example superpixel segmentation of a scene is shown in Figure 3.8.

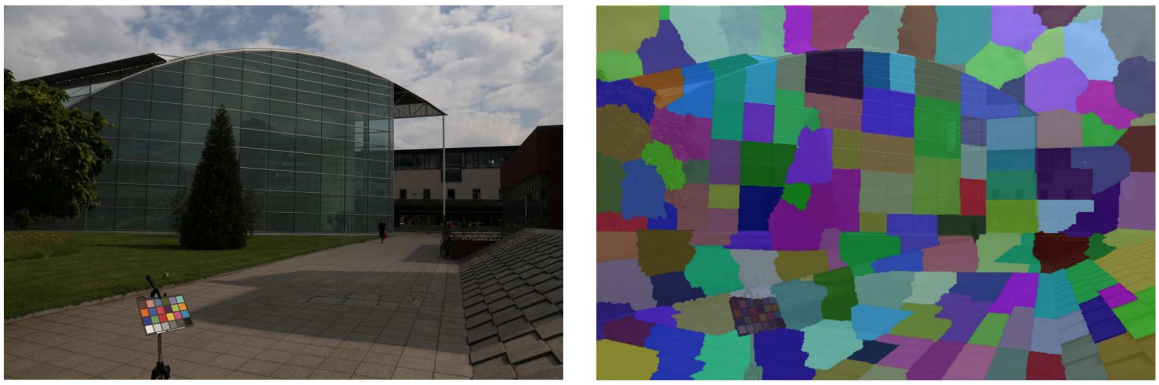

Figure 3.8: An example of Superpixel segmentation (Veksler et al., 2010). The segmentation follows edge boundaries.

The authors set up a rig where artificial scenes were made and lit by multiple filtered lamps. The scene was photographed and afterwards was spray-painted with a matte white paint. It was then rephotographed. The white images make for an approximation of the illumination map of the scene. The scenes in this dataset are shown in Figure 3.9.

The super pixel segments from the scene were matched to the illumination map, then the average of each segment was taken to be the ground truth for each superpixel in the scene. The lights in this dataset are mostly greens and yellows. This results in distinctly unnatural lighting conditions. Algorithms which make assumptions about the illuminant gamut are unlikely to perform well on this dataset.

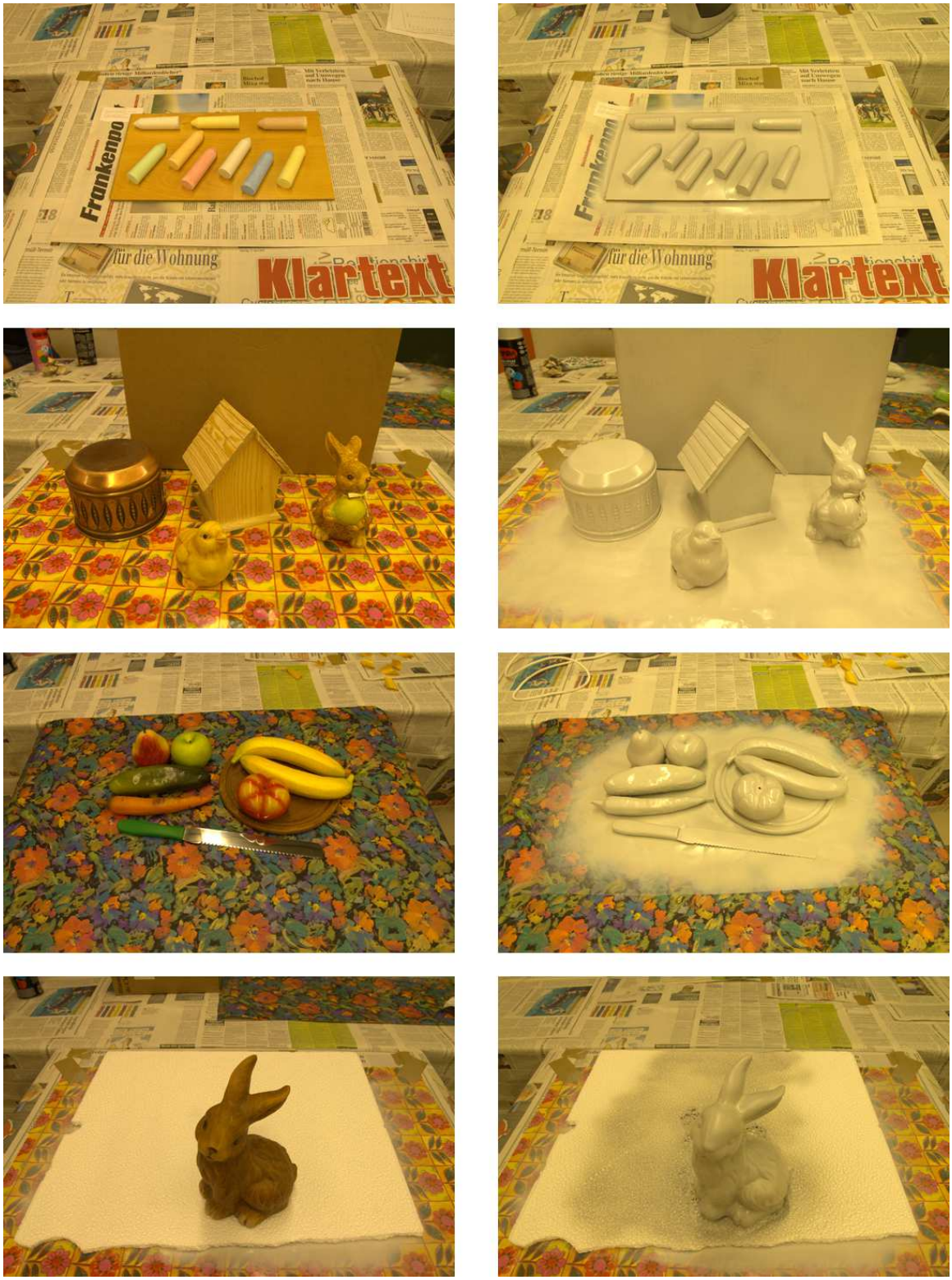

Figure 3.9: Scenes from the M Bleier et al. (2011) dataset, before and after being spray-painted white.

3.4 New Custom Shadow Datasets

In this section we will discuss two new datasets, which we been designed for two illuminant colour constancy algorithms. Both datasets contain images with prominent shadows. The first contains images of colour patches, where the second contains images of real scenes with natural multi-lit regions.

3.4.1 Colour Patch Shadow Dataset

For the first dataset, 58 images were taken using matte colour patches. Each image contains an arbitrary combination of 5 patches laid over a shadow edge. The images were taken at different times of day and at different locations in the UK using a Nikon D60 SLR camera. The patches were laid on the ground in direct sunlight, and then a block of wood laid next to them to occlude half of the light. Also, in some images artificial lighting was used to make a shadow edge.

A Macbeth Colour Chart was placed on both sides of the shadow edge, so that the 2 correct illuminant colours could be calculated. The mean of the unclipped grey patches were used as the illuminant colour. The images were rendered using the free open-source software DCRAW as linear DOC-RAW TIFFs. The simplest demosaicing algorithm was used, which was to quarter the size of the image and take the average of the two green values per pixel.

This dataset presents a particularly hard case for standard Colour Constancy algorithms, because the colour variation is small, and there are two light colours in the scene. As in the previous sections we show the distribution of illuminants in this dataset in Figure 3.10.

The distribution of lights in this dataset is fairly even with a small peak in the far red. The flat component represents the daylights captured at different times of day, and the small peak on the right represents the indoor illuminants. Even though this

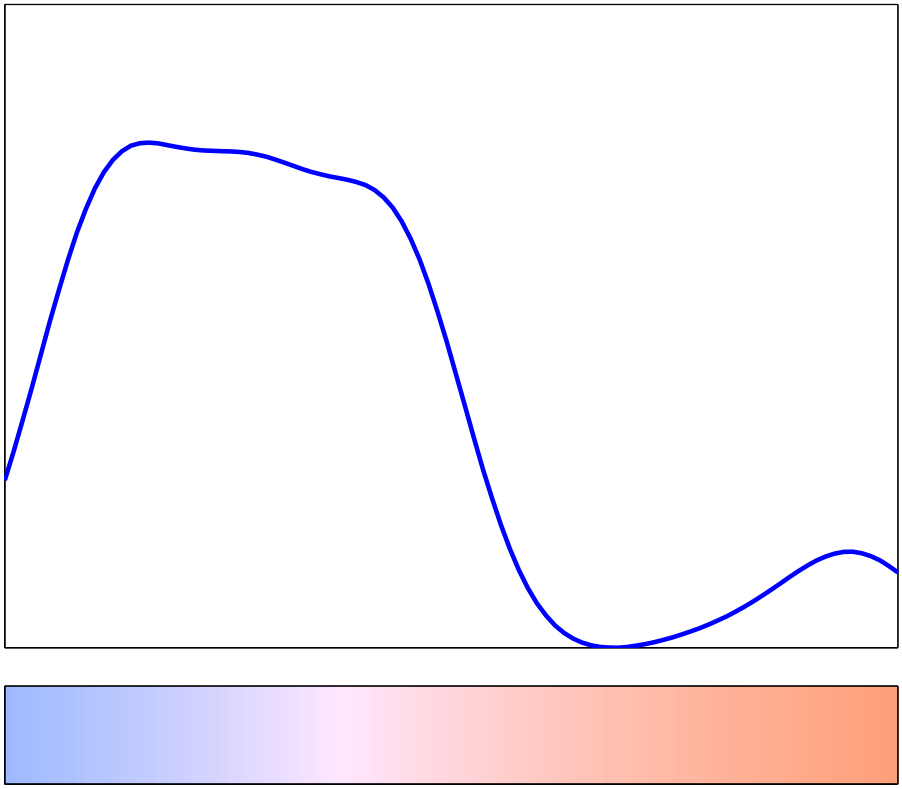

Figure 3.10: The distribution of lights in the Syntha Pulvin Colour Patches Dataset.

dataset is small compared to the other datasets, the distribution of illuminants is still wide enough to carry out representative testing. Because it is also obvious where the illumination change is, we can run traditional algorithms on this dataset as well. We can make sure that the single uniform illuminant assumption holds by only selecting patches on the same side of the shadow edge and grouping them together.

Initial Illuminant Estimation Test

Provided with this dataset is a single RGB for each colour patch under the two illuminants. We collected this by cropping out a section of each colour patch on both

sides of it's shadow edge. We then calculated the average of all the RGBs in the cropped area (to minimise noise). To conceptualise the difficulty this dataset poses to traditional algorithms, we ran several on the colours captured on each side of the shadow edge (two results per image). One of the algorithms ran was an implementation of Gamut Mapping (Forsyth, 1990). Previously, this has been evaluated as of the most successful Colour Constancy algorithms (Gijsenij et al., 2011). The algorithm requires a training phase in which a canonical gamut is generated for the associated camera sensors (refer to Section 2.5, Page 25). This is the gamut of all likely colours under a given illuminant. We generated this by photographing all of the patches in the set, shown in Figure 3.11. This may not be the ideal set for all cases, but should be favourable to this algorithm because it incorporates all the surface colours we know we will use.

Figure 3.11: All Syntha Pulvin patches under a single illuminant to make canonical colour gamut.

The angular error between the correct illuminant (taken from the white patches

on the Macbeth Colour Chart) and the estimate was taken for every iteration. The median, mean and standard deviation are shown in Table 3.1.

Algorithm	Median	Mean	Standard Deviation
GreyWorld (Buchsbaum, 1980)	13.81°	15.12°	9.06
MaxRGB (Land and McCann, 1971)	16.12°	16.22°	10.55
SoG (Finlayson and Trezzi, 2004)	15.15°	16.03°	10.15
Gamut Mapping (Forsyth, 1990)	3.19°	5.61°	4.02

Table 3.1: The results of several common Colour Constancy algorithms run on our dataset. The algorithms were run on colour patches on the same side of the shadow edge.

To make sense of these results, we refer the reader to the diagram in Figure 1.8 (Page 14). We expect an algorithm that has reasonable performance to have less than 5° angular error on average. Given the small amount of surface information as input (5 surfaces - often with little variation), it is unsurprising that the methods tested on this dataset have high angular distances. We do see a noticeable difference in the performance of Gamut Mapping though. The canonical gamut was chosen as a superset of all of the matte patches. Performance has therefore shown to be good in this case.

In Figure 3.12 we show two example images of how the dataset was captured. With a Macbeth Colour Chart placed on both sides of the shadow edge. The 'correct' illuminants can be recorded using the grey patches on these. In Figure 3.13 some of the images that are actually used for testing are shown. The images are cropped to only include the colour patches with a shadow edge.

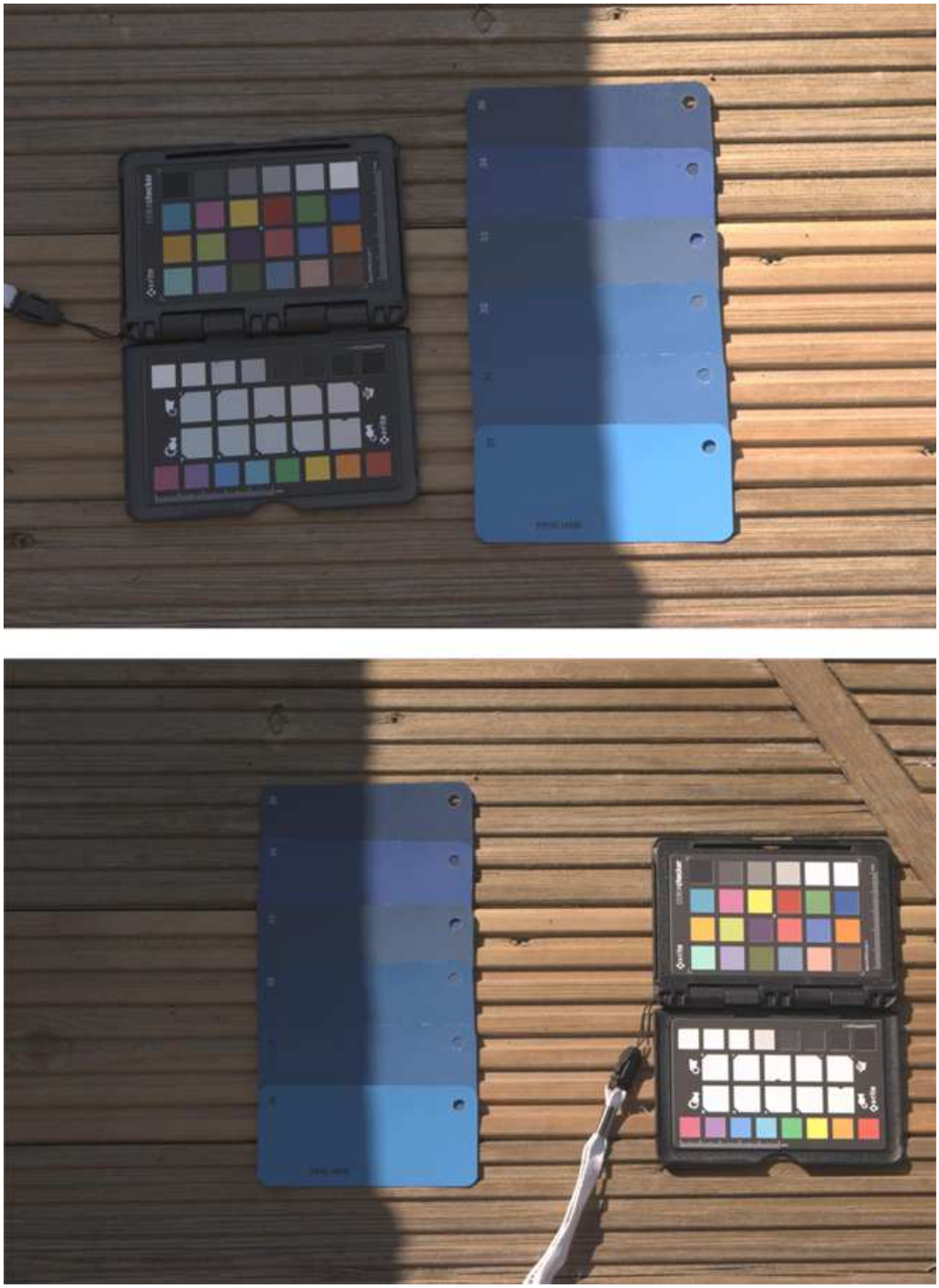

Figure 3.12: Syntha Pulvin patches with a Macbeth Colour Chart on both sides of shadow edge.

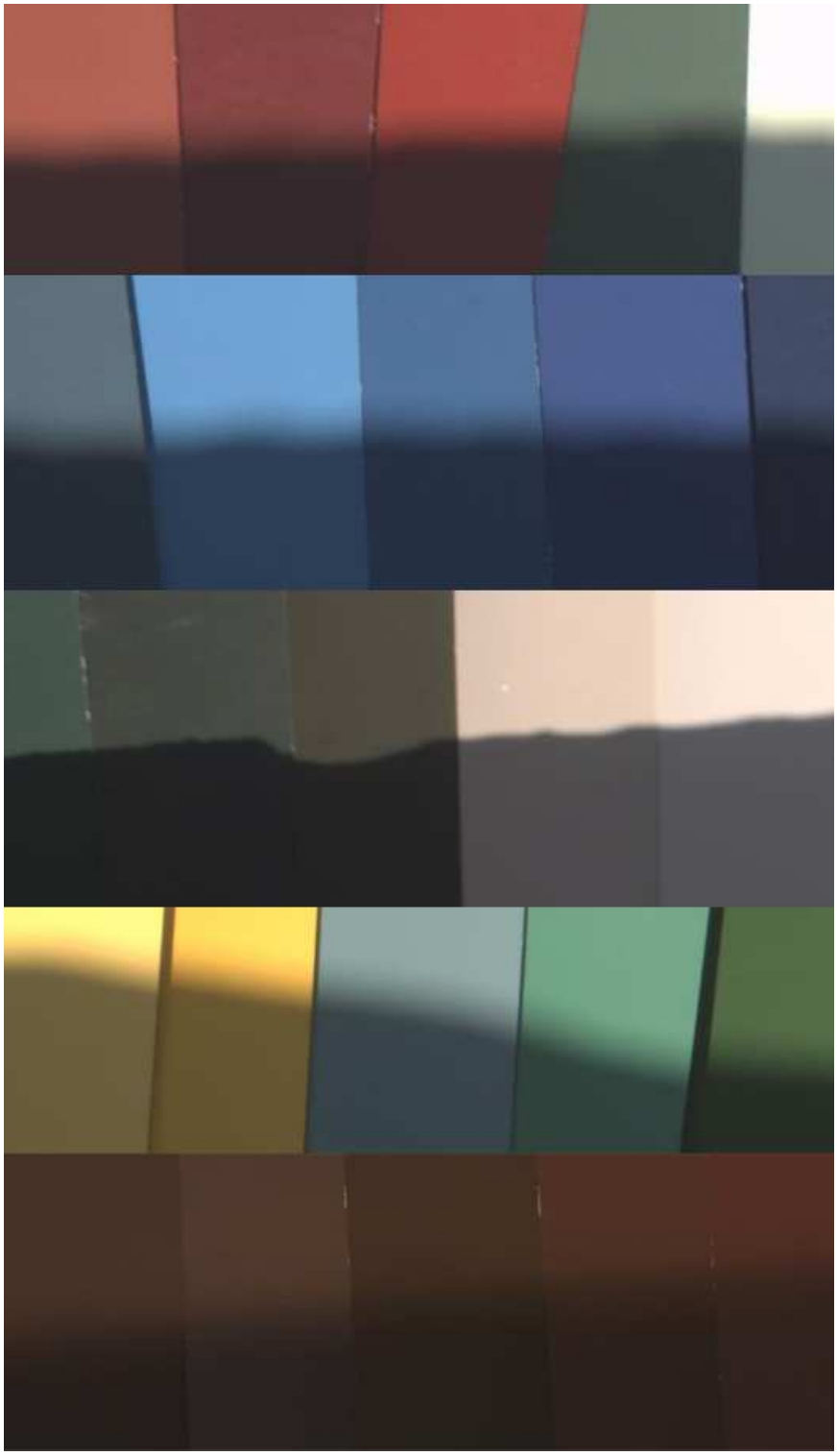

Figure 3.13: Syntha Pulvin images cropped to show only the patches with the shadow edge.

3.4.2 Shadow Images Dataset

In our second dataset, we captured real-world images that contained shadows. The dataset contains 34 images captured at different locations in the UK and in California, USA using the same Nikon D60 SLR camera. As before, after a dataset image was captured, two further images were taken of a Macbeth Colour Chart in and outside of shadow.

In Figure 3.14 we show the distribution of illuminants in this dataset. This dataset contains few chromatic reds, which are mainly found indoors or at sunset. It also contains mostly outdoor images in the daytime, and we can see the result of this in the distribution. It is important that our datasets capture a breadth of different illuminants, since varying illumination is what we are testing.

In Figure 3.15 a subset of the images in the dataset are shown. The photographs contain a variety of different scenes, all with noticeable shadows.

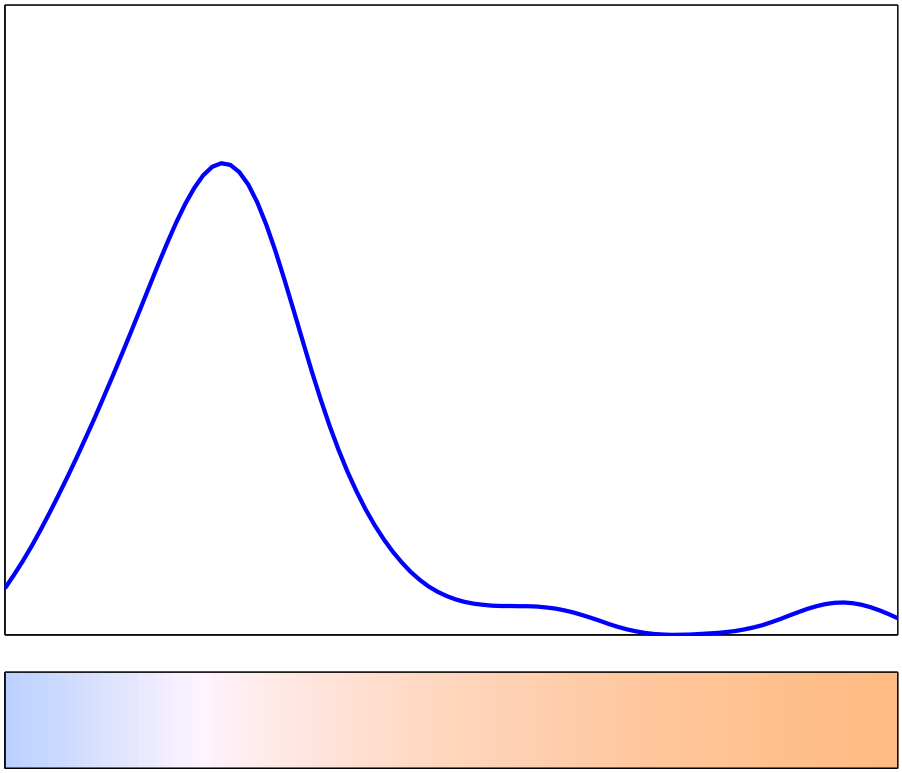

Figure 3.14: Distribution of illuminants in the shadow images dataset.

3.5 Conclusions

In this Chapter we have looked at several datasets used for Colour Constancy. We have provided a corrected rendering of the Gehler et al. (2008) dataset, which is linear and has a black level of zero. This made the images ready for immediate processing, without any additional overheads (i.e. such as running a preprocessing Matlab script).

The majority of datasets are captured with the assumption that the image contains a single uniform illuminant. Little diligence is applied to the placement of a grey

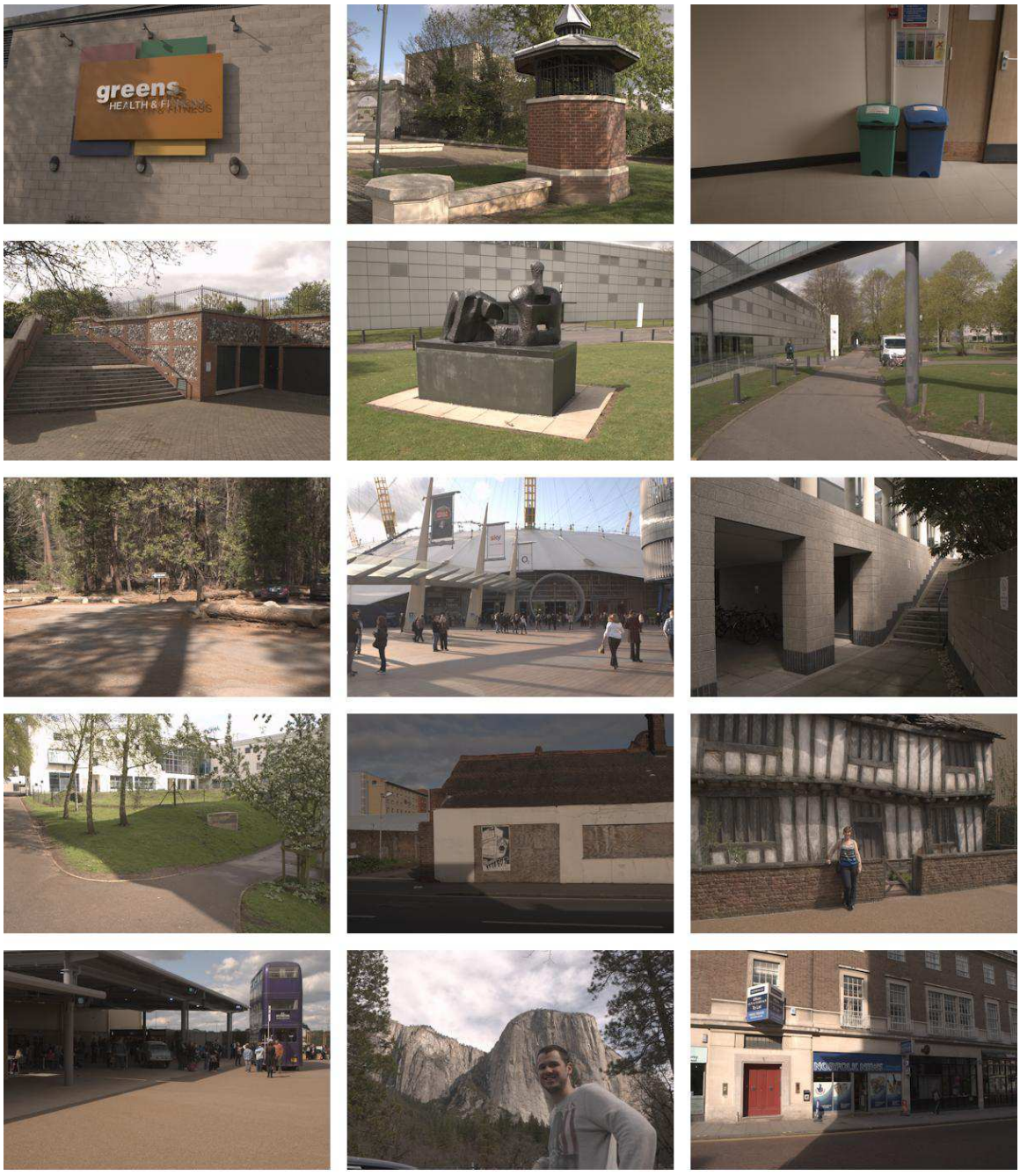

Figure 3.15: A selection of images from the shadow dataset. Each image is assumed to have two white points.

reference point in an image. Often being arbitrarily placed in and out of shadow between images. M Bleier et al. (2011) were the most diligent when designing a dataset for multiple illuminants, by actually spray painting the scene grey after capturing it under different lights. However, the usefulness of this dataset is questionable, because the illuminants were unnatural (containing green and purple lights). Works that make assumptions about realistic illuminant colours could not be rationally applied to this dataset.

We have provided two datasets which attempt to overcome the problems outlined by capturing images mostly in naturally lit scenes. All of the scenes contain shadows and additionally a Colour Checker is purposefully placed on both sides of the shadow edge. This means that the illuminant colour for the two dominant lighting fields are captured. The algorithms that are developed later in this thesis, which actively use varying illumination, require such a dataset where the lighting conditions have been properly measured.

Chapter 4

Two Illuminant Colour Constancy

4.1 Introduction

There are now several works reported in the literature which attempt to estimate surface colour when the same surface is viewed under two or more lights. There are many practical situations where such information is available including at shadow edges or where the same scene is viewed over time. Crucially, because typical lights are highly constrained, they fall on or close to the Planckian locus. Varying illumination algorithms for surface estimation can, plausibly, estimate surface colour even for scenes with little colour diversity. One of the first varying illumination algorithms made the empirical observation that the illumination mappings taking spectral band ratio chromaticities (e.g. r/b and g/b) for surfaces viewed under a range of typical lights to corresponding values under a reference canonical light (e.g. D65) tended to lie on a 2D line. It follows that applying this linear set of maps to the chromaticity of an arbitrary surface under unknown light results in a line along which the D65 counterpart should lie. Viewing the same surface under a second light produces a second constraint line. The intersection of the two lines results in an estimate of the surface chromaticity under D65. While this method can work well, Kawakami and Ikeuchi (2009) showed that serious estimation errors can result in the presence of even small amounts of image noise. While the noise tended to make only small changes

in the slope and intercept the intersection point could move a significant distance; indeed, the shifted intersection might correspond to a highly improbable (physically impossible) light. To solve this intersection stability problem they proposed limiting the set of maps not only to lie on a line but on a line segment (e.g. only allow illuminants that are physically plausible and likely). This observation, which necessitated dealing with the problem of non-intersecting line segments, formed the foundation of a new algorithm which was shown to deliver a step change in illuminant estimation. In this work we extend Kawakami and Ikeuchi (2009)'s work in two ways. First, we deal with the non-intersecting line problem using a least-squares approach (as oppose to assuming one or other of the line segments is in error). Second, we optimise over the position and length of the line segment map-set used. Experiments demonstrate that our new method delivers significantly improved illuminant estimation.

4.1.1 Background

This work is built on the work of Finlayson et al. (1995), introduced in Section 2.8.3 of Chapter 2. The authors proposed representing illumination mappings as a line in inverse-chromaticity space. We remind ourselves of the spectral-band chromaticity equation of a pixel, where the elements of $\underline{c}$ are

$$\begin{aligned} c_{r/k} &= p_R/p_k \\ c_{g/k} &= p_G/p_k \\ c_{b/k} &= p_B/p_k \end{aligned} \tag{4.1}$$

By making the blue channel the divisor we can formulate the image formation equation in chromaticity space (as in Equation 1.8). If the canonical illuminant is a uniform white light then we can write:

$$\underline{S} = (\mathbf{E}^\epsilon)^{-1}\underline{c}^\epsilon, \tag{4.2}$$

where $\underline{S}$ represents the chromaticity of the surface reflectance. In this case the diagonal elements of $\mathbf{E}^{\epsilon}$ would be the illuminant chromaticity (with their inverse making $(\mathbf{E}^{\epsilon})^{-1}$, analogous to Equation 1.4). We use Φ to denote the set of all diagonal mappings in inverse-chromaticity space, with the condition that this set is a straight line. The set $\underline{c}^{\epsilon}\Phi$ consequently describes all the possibilities for $\underline{S}$ (also a straight line). We can represent these two pixel responses as $\underline{c}^1$ and $\underline{c}^2$. Given only these two responses the authors showed that we can calculate $\underline{S}$ with the equation

$$\underline{S} = \Phi\underline{c}^1 \cap \Phi\underline{c}^2. \tag{4.3}$$

The intersection of the two mapping sets $\Phi\underline{c}^1$ and $\Phi\underline{c}^2$ will always yield a unique solution, as long as the illuminants are different. Kawakami and Ikeuchi (2009) showed that performance of this algorithm was poor on real images . They argued that this was mainly due to noise. However, any other small deviations from the model can lead to large estimation errors (e.g. inverse chromaticities don't lie exactly on a line). Finlayson et al. (1995) had shown good performance in cases where the illuminant colours were significantly different (i.e. Red and Blue).

To improve on Finlayson et al. (1995)'s approach, Kawakami and Ikeuchi (2009) proposed using a line segment instead of a full line to represent Φ. This is a more realistic representation because an infinite line includes negative chromaticities (which are not physically realisable) as well as chromaticities beyond the range of the Planckian Locus. Of course when the line segments intersect the algorithm is not different to the original Finlayson et al. (1995) algorithm. The only change is when the line segments do not intersect. This is the case when one or both of the chromaticities have values that lie outside of the range of illumination defined by the line segment.

Kawakami and Ikeuchi (2009) propose adjusting the input pixels such that the chromatic change is minimised. This adjustment is effectively a vertical or horizontal

translation of one line segment (Figure 4.1).

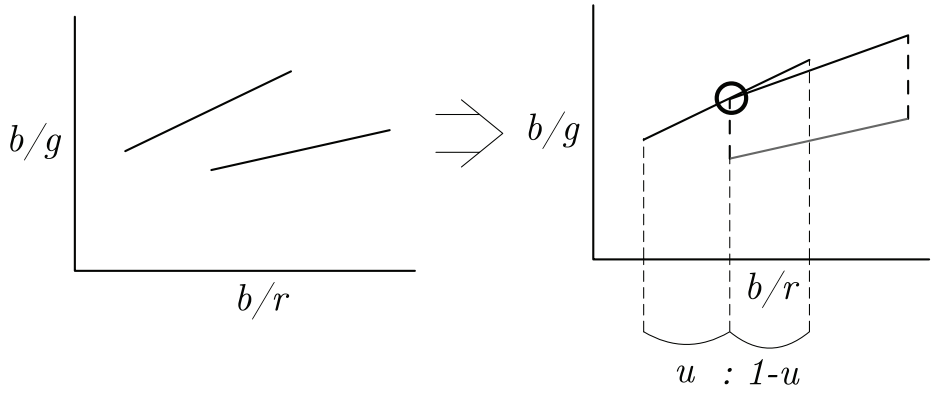

Figure 4.1: Choosing an intersection point of non-intersecting mapping line segments (Kawakami and Ikeuchi, 2009).

4.2 The Least-Squares Optimal Intersection

In the previous section we learned that modelling illumination mappings as a line segment can help stabilise surface colour estimation. The line segment is an attempt at stabilising the error that using an infinite line can cause. The algorithm forces its output mappings to both lie exactly on the line segment. In this section we propose a new method for choosing an intersection point, that is least-squares optimal.

4.2.1 Intersecting Line Segments

Here we wish to revisit how we find the point which best approximates the intersection (of non-intersecting line segments). In Figure 4.2 we show two non - intersecting line segments. We ask which point (anywhere on the Cartesian plane) is simultaneously closest to both segments. Let us denote the line segments as $l1$ and $l2$. Abstractly, we might write our objective function as

$$\min_{p} \text{dist}(p, l_1) + \text{dist}(p, l_2), \tag{4.4}$$

where $\text{dist}(p, l)$ is the shortest length between point p and line l. Clearly, the distance to the line segments is defined by a perpendicular projection. Such projections for the endpoints of each line to the other line segment are shown in Figure 4.2. Let us now consider where the actual intersection point might be. Could it be, for example, between the endpoints for both lines? This is impossible. The easy case is shown in Figure 4.4 where the ranges of the x-coordinates of the line segments do not overlap. Here the best intersection is the middle of the two closest endpoints. In the second case we assume the range of the x-coordinates do not overlap. Here as a function of the x-coordinate and barring parallel lines the two segments will clearly converge in one direction (get closer) and diverge in the other. They converge in the left direction of Figures 4.2 and 4.3. Clearly, as we move an intersection point in the converging direction the length to both lines (their respective perpendicular projections) will decrease. Of course we can move the point only so far until the closest point on either line segment is the endpoint of one of the lines. Thus, we find the intersection point by calculating the closest point to each endpoint in the other line segment (Figure 4.4). We find the projection point with the minimum distance. The intersection point is simply the average of the two points. All computation is in closed-form.

One advantage of our intersection problem is that unlike the Kawakami and Ikeuchi (2009) method, we always have a well defined intersection point (least-squares optimal). This method does not restrict the solution to lie on the line segment, but finds a point which best fits the data. We summarise the algorithm to find the point closest to both lines as:

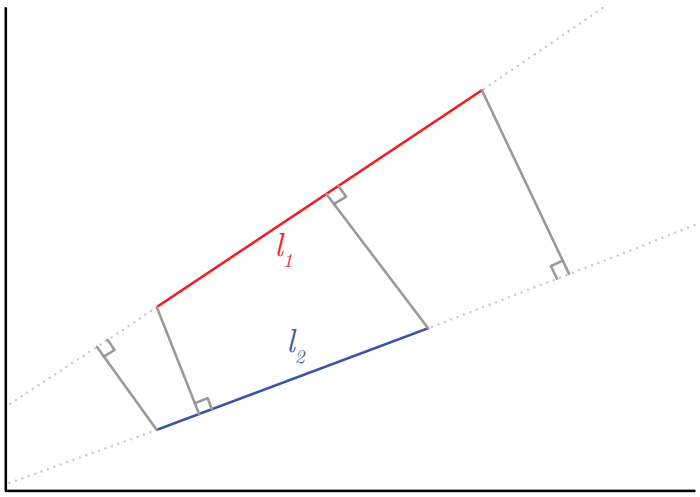

Figure 4.2: Perpendicular projections of the line segment end points onto the extended full line of the other line segment.

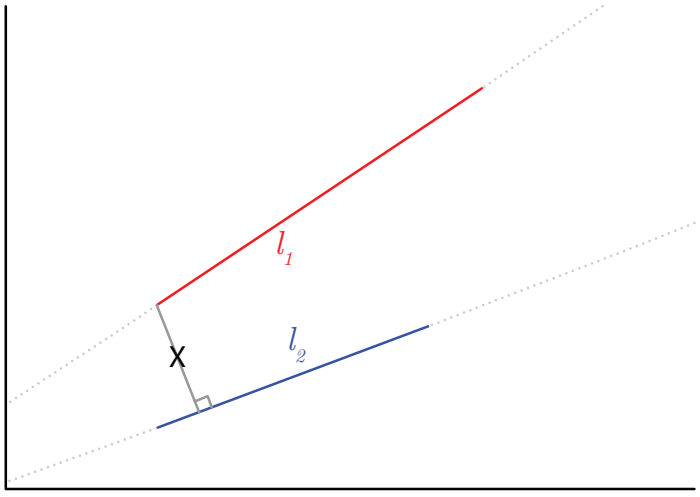

Figure 4.3: Perpendicular projection with point closest to both lines marked.

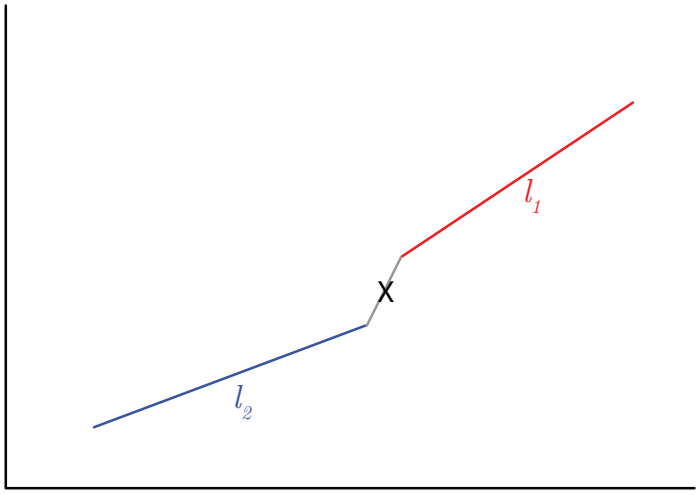

Figure 4.4: Edge case where Perpendicular Projection doesn't produce a valid distance line. The new distance line is drawn between the line segment edge points.

1. **If** Line Segments intersect, return intersection point.
2. **Else** solve for all the projections to the other line segment (4 projections as shown in Figure 4.2)
3. Remove any points found that do not lie on the line segment.
4. **If** Number of resulting points > 0, choose the point that projects with the shortest distance. Return point half-way along this projection (Figure 4.3).
5. **Else** return the halfway point on the line of shortest distance between end points of the two segments (Figure 4.4).

The point returned is the estimate for the chromaticity of the surface reflectance $\underline{S}$. We can convert this to an estimate of the two illuminants by following the diagonal model. We shall call the jth illuminant estimate $\underline{E}^j$ and it is calculated as $\underline{E}^j = \underline{c}^j / \underline{S}$, where the division is a vector point-wise division.

Of course, the intersection point we calculate will depend on the line segment used to model the illuminant maps. Indeed, if we made the segment longer and longer we would eventually observe the same behaviour as the original Finlayson et al. (1995) algorithm: as the segment grows in length the non-intersection problem would become less frequent. In the opposite direction if we assumed the illuminant map to be a single map (i.e. the identity transform) then the resulting constancy algorithm resorts to averaging the two input chromaticities (a sort of Grey World approach).

4.2.2 Calibrating The Line Segment

The second innovation is to consider algorithm performance as a function of line segment length, position and orientation. The mapping line segment is intricately linked to the sensors of the photo system. We predict using a slightly shorter line compared with the physical limits implied by Planckian illumination will result in improved performance. The most saturated lights in our model appear less frequently than those closer to white. By shortening the line segment representing the set of light maps we will perform slightly less well for strongly chromatic light but better on average.

To calibrate the line segment, a selection of surfaces under different lights is needed. We can define a data store $\Omega_{i,j}$, where the ith elements represent the surface RGBs under the jth illuminant. A line segment can be represented as a point in 4 dimensional space $\underline{L} = \{m, c, x_0, x_1\}$. We can represent the line segment colour constancy algorithm as a function which takes three inputs $(\Omega_{x,1}, \Omega_{x,2}, \underline{L})$. The algorithm outputs two illuminant estimates $\{\underline{\epsilon}^1, \underline{\epsilon}^2\}$. We also know the pair $\underline{E}^1, \underline{E}^2$ which are the correct illuminants. For each pair of lights and for each surface we can calculate a pair of angular errors.

We define the total error of each iteration as angular$(\underline{\epsilon}^1, \underline{E}^1)$ + angular$(\underline{\epsilon}^2, \underline{E}^2)$.

The sum of these errors for every combination in our data store, can be thought of as the total error associated with the line segment $\underline{L}$. We would like to find a line segment $\underline{L}$ that minimises this error. This is clearly a non-linear minimisation. The most straight forward way to find a good answer is to search. We search for the line segment $\underline{L}$ that minimises this total error for a given representative data store. By picking a sensible starting position we can be confident that the search will optimise the algorithm.

Choosing a Starting Position

By capturing a Macbeth Colour chart we can recover the RGB of the illuminant by using the RGB of the grey patches (i.e. taking the mean of the non-clipped RGBs). As suggested by Finlayson et al. (1995) we can use linear regression to fit a line to the data. The line endpoints can be chosen as the points at the two extremes of the data. This means the starting line segment covers the full breadth of our input illuminants. We use the function minimisation technique described by Lagarias et al. (1998). This is implemented in MATLAB as the function fminsearch.

We chose 5 surfaces from our dataset that varied in chromaticity. We populated the datastore with surfaces under two thirds of the available illuminants. We could then test on the other surfaces under the remaining third of illuminants. This provides cross-validation, as so does not introduce bias by testing on trained data.

4.2.3 Modified Kawakami and Ikeuchi (2009)

The last innovation is to apply the search step from the previous section to the Kawakami and Ikeuchi (2009) algorithm. It's current calibration step involves choosing a line that falls between two temperatures on the Planckian Locus. Even though the Planckian Locus appears to be a good model for illumination change, it does not guarantee the best choice of line segment. However, the Kawakami and Ikeuchi

(2009) algorithm isn't complete because there are cases that do not provide a solution (i.e. when the line segments do not overlap in either axis). In this circumstance we cannot get an angular distance for the given input. We consider this to be equal to the largest error the algorithm could make, which would be 90° (i.e. estimating that the RGB is Red when it is actually Blue). This means that the search will optimise for line segments that actually output a solution.

4.3 Experiments

In this section we look to examine the performance of our line segment algorithm as well as the modified Kawakami and Ikeuchi (2009) algorithm on synthetic data and real data, and benchmark it against other work.

4.3.1 Synthetic Data

Here we look to analyse the performance on synthetically generated data. The benefit of using synthetic data is that we have complete control, and the data is free of noise. We also know that the rendering exactly conforms to the image formation model. This gives an idea of how the algorithm will perform in ideal circumstances. We use the set of 99 daylight spectra recorded in Granada, Spain by Hernández-Andrés et al. (2001). For the surfaces, we use the set of spectral reflectances gathered by Barnard et al. (2002), along with the Sony DXC-930 camera sensors also provided. The algorithm requires a training phase to select a suitable line segment for the camera used. We used the method of cross-validation for training. This involves splitting up the surfaces under all illuminants randomly into thirds. We shall train on two-thirds, by searching for a good line segment (Section). We will then test on the remaining third. This is cycled three times so that all the data is evaluated (this is the same methodology prescribed by Gijsenij et al. (2011)). The data store used to train is

gathered using RGBs generated from 5 surface reflectances, under two-thirds of the illuminant set. The optimal line segment is output from this process. The algorithm is then tested on 5 different surfaces rendered under the remaining 33 illuminants.

To benchmark our algorithm we compare it against the Finlayson et al. (1995) algorithm and the Kawakami and Ikeuchi (2009) algorithm. We also apply our modified version of the Kawakami and Ikeuchi (2009) algorithm, which includes our trained line segment (Section 4.2.3). We show the mean, median and standard deviation of these measurements in Table 4.1

Algorithm	Median	Mean	Standard Deviation
Finlayson et al. (1995)	22.20°	36.64°	35.208
Kawakami and Ikeuchi (2009)	8.90°	11.93°	55.79
Kawakami and Ikeuchi (2009) (Modified)	3.46°	5.04°	4.28
Our Least-Squares Algorithm	2.86°	3.74°	3.02

Table 4.1: Table comparing the angular errors of 2 illuminant 1 surface algorithms to our Least-Squares algorithm, on synthetic data.

The results show that the Finlayson et al. (1995) algorithm has particularly poor performance. However, the authors stated that performance is improved when illuminants are dissimilar, to maximise the angles between the two lines. This was not accounted for in the experimentation and so resulted in poor performance. The Kawakami and Ikeuchi (2009) algorithm performs significantly better. However it's performance is still unacceptable on average (larger than 5° error. We were able to drastically improve the performance of this algorithm simply by searching for an optimal line segment (Section 4.2.3). We highlight that the search optimisation was successful in choosing a line segment which always returned an answer (this was particularly good as output of the Kawakami and Ikeuchi (2009) algorithm is undefined for some cases). We can improve performance even further by using our own least-squares algorithm. Not only is it's average result the lowest in angular error, but it

also has the lowest standard deviation. This is remarkable given the small amount of information required as input for this algorithm.

4.3.2 Colour Chart Dataset

Synthetic data is a good way to test an algorithm in ideal circumstances. However, it is not an exact representation of the data that could be captured using real equipment. In this section our algorithm will be tested using 482 images taken from the Gehler et al. (2008) dataset. These were captured on the Canon 5D DLSR camera. We use our own linear rendering of this dataset (as described in Section 3.3.3, Page 53). The dataset contains a Macbeth Colour Chart in every image, which means that there are 24 surfaces captured under 482 lights. Using the coordinates provided by the authors, the pixels of each patch of the Macbeth Colour Chart were cropped out of the images. The average was taken to get an accurate reading of each patch with minimal noise. The experimentation is constructed in the same way as the previous section. We use three-fold cross validation to train the algorithm and generate a mapping line segment. Five surfaces were selected from the Colour Chart to train on, and a different 5 surfaces were tested on. Given that 6 of the patches are achromatic, the 5 surfaces were not chosen randomly, but manually. This was to ensure that the test set was broad for training. Unlike the previous section, which only used natural daylights, this dataset also incorporates some indoor fluorescent lighting. The correct illuminant was taken as the average of the non-clipped achromatic patches (i.e. patches that were not [255 255 255]) on the Colour Chart. One again, since we cannot recover brightness information then we use the angular distance between the correct illuminant and the estimated illuminant to measure performance. The results are shown in Table 4.2.

On real data we see the same trend as in Table 4.1. The Finlayson et al. (1995)

Algorithm	Median	Mean	Standard Deviation
Finlayson et al. (1995)	12.58°	25.30°	33.47
Kawakami and Ikeuchi (2009)	11.99°	15.10°	16.09
Kawakami and Ikeuchi (2009) (Modified)	4.47°	10.02°	12.13
Our Least-Squares Algorithm	4.31°	6.29°	5.08

Table 4.2: Table comparing the angular errors of 2 illuminant 1 surface algorithms to our Least-Squares algorithm, on the Gehler et al. (2008) dataset.

algorithm performs least well, with a large standard deviation. Again, it should be noted that this algorithm performs better in cases when the lights are significantly different. We are able to improve on the Kawakami and Ikeuchi (2009) algorithm drastically by doing the search-based optimisation. However, on real data, the algorithm did fail to return an answer in 1% of cases. Compared with our own algorithm, which returns an answer in all cases, the Kawakami and Ikeuchi (2009) algorithm still does not perform as well over all. To demonstrate the benefit of testing pairs of lights that are significantly different, we run the same set of experiments again but reduce the illuminant pairs we use. Instead of using all illuminant pairs, we use only the pairs where the angular distance between the illuminants is more than 3 degrees. We show the result of this in Table 4.3.

Algorithm	Median	Mean	Standard Deviation
Finlayson et al. (1995)	12.05°	14.15°	14.43
Kawakami and Ikeuchi (2009)	5.43°	7.93°	5.70
Kawakami and Ikeuchi (2009) (Modified)	3.61°	8.96°	5.31
Our Least-Squares Algorithm	3.51°	4.81°	4.19

Table 4.3: Table comparing the angular errors of 2 illuminant 1 surface algorithms to our Least-Squares algorithm, on the Gehler et al. (2008) dataset. Only pairs with an angular distance of more than 3 degrees were using in training and testing.

By constraining the light pairs, the performance of the algorithm is improved.

The Finlayson et al. (1995) algorithm shows a large improvement, particularly with the smaller standard deviation. This shows a reduction in the extreme errors that the algorithm could output. However, it's performance is still not in the ballpark of the other algorithms. Emphasising the importance of restricting the mapping set.

4.3.3 Colour Patch Shadow Dataset

In Section 3.4.1 we introduced a new dataset containing shadows. Each image contains 5 matte, coloured patches that have been placed on a shadow edge. This dataset provides us with the types of illumination change that we could encounter in the real world. The results are shown in Table 4.4.

Algorithm	Median	Mean	Standard Deviation
Finlayson et al. (1995)	16.52°	33.76°	36.79
Kawakami and Ikeuchi (2009)	9.28°	11.27°	8.61
Kawakami and Ikeuchi (2009) (Modified)	8.76°	10.76°	9.18
Our Least-Squares Algorithm	3.41°	4.61°	3.60

Table 4.4: Table comparing the angular errors of 2 illuminant 1 surface algorithms to our Least-Squares algorithm, on the Colour Patch Shadow Dataset (Section 3.4.1).

The Kawakami and Ikeuchi (2009) method performs quite poorly, even with our modified version. For comparison, in Table 3.1 in Chapter 3 (Page 44) we can see our algorithm significantly outperforms some other traditional colour constancy algorithms (which did not produce an average error less than 13°). These algorithms only used the small amount of varying surface information in the images.

4.4 Conclusions

In this Chapter we have looked at Two Illuminant Colour Constancy. We used the assumption of Kawakami and Ikeuchi (2009) that all illuminant mappings lie on a line segment in chromaticity space. We can multiply two pixel chromaticities of the same

surface under two lights by this line segment. This will produce two line segments which are estimates for the surface reflectance chromaticity. If the line segments intersect then this is our single solution (the same as the Finlayson et al. (1995) method). If they do not intersect we proposed using the point simultaneously closest to both line segments. This method had better performance than previous work. We also proposed a mechanism for choosing the illuminant mapping line segment by searching. By applying this to the Kawakami and Ikeuchi (2009) algorithm we were also able to improve their result, however it still did not perform as well as ours.

The majority of Colour Constancy algorithms assume that an image contains many surfaces under a single illuminant. Therefore, they put a lot of focus on using only varying surface information. The results we present here are important for two reasons. First, images with two lights are notoriously hard (often impossible) to solve for the majority of algorithms which assume there is a single illuminant present. Second, even when there is just a single surface under two different lights, we get a strong cue to what the illuminant colour actually is. Gamut Mapping (Forsyth, 1990), for instance, requires a lot of surfaces to get good performance. For most images this is usually not a problem. However, because of the predictability of illuminant colours, a much smaller amount of information is needed to make our algorithm work. Recent research in the field has not been trying to solve the whole problem of colour constancy. But rather focus on specific cases that it can solve well. This work is an example of that trend in that we are not proposing a complete solution to Colour Constancy, but rather, have identified a particular case which we can solve (and for which traditional algorithms cannot).

We have shown that a relatively simple training step in choosing the mapping line segment can dramatically increase performance, not only in our own algorithm but also in our predecessor's work. Our algorithm also has the advantage of outputting

a well-defined solution in every circumstance (assuming both input illuminants are indeed different). This is in contrast to the previous work, which had a failure case (the authors did not discuss this in any detail). Although, it is noted that in only a very small amount of cases did their algorithm actually fail to provide answer. Given that our algorithm provides a simple change, which results in significantly better performance on average, there is unlikely a reason to use the predecessor versions over ours in future. In later chapters we will talk about how we have developed this work to include surface information, and how we can automatically find shadow edges in images.

Chapter 5

Unifying Surface and Illuminant Constraints for Improved Illuminant Estimation

5.1 Introduction

As surfaces vary throughout a scene, the reflected light also changes. Similarly changes to the illuminant also affect reflected light. DZmura and Iverson (1993a) were interested in modelling this relationship using bilinear models, and finding the scenarios where colour constancy had a closed-form solution. The cases where a solution was possible were not well-matched to real-world scenarios (i.e. requiring too tight constraints on the number of illuminants in a scene and how surface reflectances vary). However, the work showed that varying illumination as well as varying surface information relaxed these constraints and allowed for colour constancy to be solved in broader circumstances.

In this work we are also interested in the relationship between varying surfaces and varying illumination in images. Illuminant colours in the world are naturally constrained and, to a certain extent, predictable. We have previously shown this by modelling all illuminants as a line segment in inverse chromaticity space. However, the model is only an approximation, and we are now interested in broadening how

we represent real-world illuminants.

Forsyth (1990) developed one of the most successful colour constancy algorithms (Gamut Mapping) by making the simple observation that surface colours are constrained by the colour of the illuminant. From this simple observation he generated an algorithm which outputs a set of possibilities for illuminants, given the surface colours of an image. The more surface colours present in a scene, the more illuminant colours that could be ruled out. Finlayson (1996) extended this approach by simply constraining the set of possible illuminants to only the ones that had been previously observed in the real world. For example, the Gamut Mapping algorithm may suggest that the illuminant could be green or purple. This is obviously possible, but very unlikely in most real-world scenarios. Particularly in natural scenes.

In this work we unify the line segment intersection method with Gamut Mapping, to account for the relationship between different numbers of illuminants and surfaces in a scene. The locations of surfaces under different illuminants would have to be segmented and extracted for this process to be fully automated, which we do not focus on in this work. However, our results show that the performance of traditional gamut mapping can be significantly improved when modelled in this way.

5.1.1 Background

Historically in illuminant estimation, sparse literature has focused on the interrelationship between the illuminants and surfaces in a scene. However, DZmura and Iverson (1993a) investigated this relationship. They extended the work of Maloney and Wandell (1986) by furthering their finite-dimensional model of surface reflectance to incorporate an additional illuminant constraint. By introducing more than one illuminant, the strict assumptions about the variance of light and surface spectra could be relaxed. They showed that in a tristimulus system, where surface and illuminant

spectra were modelled by 3 respective basis functions, colour constancy was theoretically possible. We previously explained that surface spectra cannot be well modelled by only 3-degrees of freedom. This alone means that the method has little practical value.

Subsequent work has supported the idea that the varying surface reflectances in a scene naturally constrain the colours that could have illuminated them (Forsyth, 1990). We have also learnt that varying illuminants in a scene provide a valuable constraint in surface reflectance estimation (Finlayson et al., 1995; Kawakami and Ikeuchi, 2009). In this work we unify these two ideas into a single framework, to output the minimum solution set for the illuminant colours in a scene.

5.1.2 Revisiting Gamut Mapping

In Section 2.5 (Page 25), we introduced the Gamut Mapping algorithm. We defined the points on the convex hull of a set of representative surfaces under a canonical illuminant as $\Gamma(\mathcal{P})_i$ (where i indexes the i^{th} hull point). A mapping which takes pixel $\underline{p}$ from some unknown light into this set can therefore be defined as $\mathcal{D}_i = \Gamma(\mathcal{P})_i/\underline{p}$. We can map an entire image of pixels $\mathcal{I}$ (simply represented by it's convex hull points $\Gamma(\mathcal{I})_j$) into $\mathcal{P}$ by intersecting all of the resulting mapping sets $\mathcal{E}_j$. Ending with the final set of plausible mappings

$$\mathcal{M} = \bigcap_{j=1}^{n} \mathcal{E}_j, \tag{5.1}$$

where n is the number of vertices required to define the convex set $\mathcal{I}$, which is subsequently the number of mapping sets in $\mathcal{E}$. Assuming that the canonical illuminant is a uniform white light, then the inverse of $\mathcal{M}$ is the set of possibilities for the illuminant RGB.

This methodology has also been shown to work as well in $(r/b, g/b)$ chromaticity

space (Finlayson, 1996). A colour constancy algorithm using this space will recover only the illuminant mapping chromaticity. In Figure 5.1 the gamut mapping algorithm is visualised in 2D chromaticity space. The steps of the algorithm can be applied straight from the RGB implementation to 2D chromatcity space.

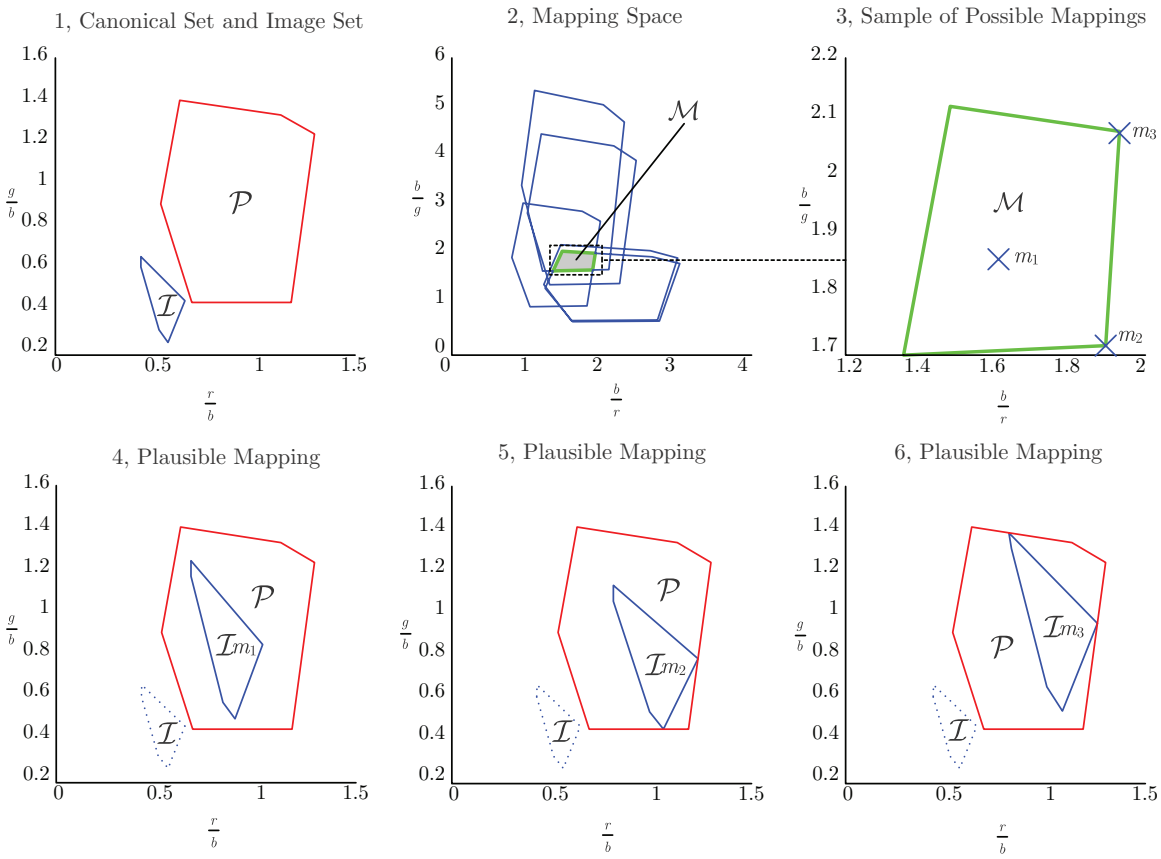

Figure 5.1: **Top row:** shows, in 2 dimensions, how a solution mapping set $\mathcal{M}$ can be obtained for an image set $\mathcal{I}$ and canonical set $\mathcal{P}$. **Bottom Row:** Three example plausible mappings as sampled from the set $\mathcal{M}$, where $\{m_1, m_2, m_3\} \subset \mathcal{M}$.

Additionally in his work Finlayson (1996) introduced an illuminant mapping constraint to the algorithm. Given a set of plausible illuminant RGBs it is possible to build a convex set (i.e. a gamut) from them. Each point in the convex set represents a potential scene illuminant. This represents the same set as $\mathcal{M}$. Let's call our convex set of illuminants $\mathcal{L}$. We can extend equation 5.1 to become

$$\mathcal{M} = \left(\bigcap_{j=1}^{n} \mathcal{E}_j \right) \bigcap \mathcal{L}^{-1}. \tag{5.2}$$

This unifies the surface constraints with an illuminant constraint. Note that the inverse of the convex set $\mathcal{L}$ will no longer be a convex set. This means that $\mathcal{M}$ may also not be convex. For ease of implementation we can take the convex closure of $\mathcal{L}^{-1}$ to preserve convexity. This reduces the computational complexity, but it will broaden the set of plausible illuminants.

Finlayson and Hordley (1998) investigated the problem of choosing a single mapping from $\mathcal{M}$. They argued that every mapping in the set was equally likely. From a statistical point of view, they choose the point that would minimise the error on average. This means choosing the centroid of the convex set. The set $\mathcal{M}$ in chromaticity space lies on the plane that goes through $b = 1$. The set becomes a conical mapping set in RGB space. We want to find the centroid of points in this shape that have the same intensity (i.e. a fixed magnitude). We project this set in chromaticity space onto the surface of a sphere in RGB space. The authors proposed sampling within the shape randomly, and taking the mean of the points.

5.2 Unifying Illuminant Mapping Constraints with Gamut Mapping

For this work, rather than taking all the pixels in an image we propose segmenting it into areas of the same illuminant colour, we could then apply Gamut Mapping separately to those areas. This is a sensible way of applying Gamut Mapping, because the input data would accurately reflect the model the algorithm relies on. Segmenting images into areas of uniform illumination is a difficult problem, and is beyond the scope of the work. However some research has been done which attempts to address this issue, such as the work of Finlayson et al. (2007) which uses the differences

between a colour filtered and non-filtered image to detect changes in illumination. The framework which we have designed aims to fully use the information in a scene with multiple illuminants to recover estimates for those illuminants, by only using gamut constraints. We take what we have learned from our previous work, to apply stronger constraints than just running Gamut Mapping on the areas in a scene of uniform illumination.

Consider an image taken of a scene in an outdoor scenario on a sunny day, with cast shadows (such as the one shown in Figure 5.2). The traditional gamut mapping algorithm would attempt to find a single white point. Indeed, we know this is often the goal of camera white balance. We know this goal, under this circumstance, is fundamentally flawed. The surface reflectances are affected by the ambient daylight and that of direct sunlight. In terms of chromaticity, we know the two to be significantly different. Therefore assuming all the pixels in the scene form a representative gamut of surfaces under a single illuminant is nearly always erroneous.

Figure 5.2: An example segmentation of areas of uniform illumination in an image (i.e. shadow/non-shadow). Areas have been blacked out to highlight the different segments.

Let's now assume we were able to segment this scene into areas of uniform illumination. We assume this image contains two illuminants, shadow and non-shadow. However, this model can be generalised to n^L illuminants. From this segmentation

we can create n^L chromaticity gamuts (one for the surfaces under each light). We have represented this type of gamut for an entire scene before as $\mathcal{I}$. In equation 5.2 we defined the mapping set for surfaces under a single illuminant. We generalise this for the L^{th} illuminant as

$$\mathcal{M}_L = \left(\bigcap_{j=1}^{n} \mathcal{E}_{L,j} \right) \bigcap \mathcal{L}^{-1}, \tag{5.3}$$

where $\mathcal{M}_L$ describes the mapping set from surfaces under the L^{th} scene illuminant to the uniform white illuminant. In most scenes there is some cross over from one illuminant to another on the same surface. We can see this in Figure 5.2 at the shadow boundary, where the orange colour of the brick on the building is captured in shadow and out of shadow. We define $z \subset x$, which is the surfaces in a scene that appear under more than one illuminant. We can use the mapping sets described in equation 5.3, to solve for the set of possibilities for the surface chromaticity. This is an extension of the ideas outlined in Chapter 4, where pixel chromaticities are multiplied by mapping sets, and the resulting intersection contains the surface chromaticity

$$\mathcal{S}^z = \bigcap_{L=1}^{n^L} \underline{p}_L^z \mathcal{M}_L, \tag{5.4}$$

where n^L is the number of illuminants. We now have a set of canonical possibilities for each surface illuminated by more than one light in the scene. Finally, we can solve for the chromaticities of each of the scene illuminants. We can think of $\mathcal{D}_L^{-1}$ as the mapping which takes a pixel response chromaticity to a surface reflectance chromaticity. The mapping can only be valid if it takes $\underline{p}_L^z$ into the set $\mathcal{S}^z$ for all n^z surfaces. The mappings must fall within the following set

$${\mathcal{D}_L}^{-1} = \bigcap_{z=1}^{n^z} \left(\frac{\underline{p}_L^z}{\mathcal{S}^z} \right). \tag{5.5}$$

The size of this set is affected by the broadness of the surface gamut, and the chromaticity difference of the scene illuminants. Notice that $\mathcal{D}^{L^{-1}} \subset \mathcal{D}^{-1}$. The algorithm can shrink the size of the set, compared to doing Gamut Mapping alone. This provides a unifying framework where we can use information in a scene lit by multiple illuminants. We summarise our generic framework below:

1. Collect a set of surface chromaticities $\mathcal{P}$ under a uniform white illuminant.

2. Collect a set of representative illuminant chromaticities $\mathcal{L}$. The inverse of this set $\mathcal{L}^{-1}$ can be thought of as illuminant mappings.

3. A given input image contains n^L illuminants and x surfaces, with potentially some surfaces under all illuminants.

4. For groups of surfaces under the same illuminant we can perform traditional Gamut Mapping to gain n estimates for $\mathcal{M}_L$.

5. Each set should only contain mappings that are in $\mathcal{L}^{-1}$, so we intersect each with our output mapping set from Gamut Mapping (Equation 5.3).

6. For a given surface under all n^L illuminants, we can multiply each surface chromaticity by it's associate mapping set $\mathcal{M}_L$, and do the intersection to estimate $\mathcal{S}^z$ (Equation 5.4).

7. We can then convert this surface estimation set into illuminant estimate sets (Equation 5.5).

8. Finally we choose an illuminant in each set following the methodology described by Finlayson and Hordley (1998).

It is important to note that in the case of more than two illuminants, we do not necessarily require the same surface under all the illuminants for the algorithm to

function. We can derive $\mathcal{S}^z$ from the information we have available. If the scene only contains the same surface under a maximum of two illuminants, but the scene is lit by more than two, we can just apply equation 5.4 to those surfaces. We are able to tailor the constraints to the image.

Currently the algorithm relies on the assumption that the sets to be intersected will always overlap. Practically this is not the case, and to always guarantee that the algorithm can provide a sensible solution, it is important to deal with non-intersection. We review this in the next subsection.

5.2.1 The Optimal Intersection of Nonintersecting Convex Sets

Let $\mathcal{W}_i$ be the set of points that lie in the i^{th} convex set and $\underline{q}$ be the point (a, b) which is simultaneously closest to all sets. To solve this problem we need to minimise the function

$$\min_{\underline{q}} \sum_{i=1}^{m} \text{dist}(\mathcal{W}_i, \underline{q})^2, \tag{5.6}$$

where the function $\text{dist}(\mathcal{W}_i, \underline{q})$ returns the Euclidean distance between point $\underline{q}$ and the set $\mathcal{W}_i$.

We are going to solve for $\underline{q}$ iteratively. In our iterative algorithm let $\underline{q}^j$ denote the best-guess for the optimal intersection at iteration j. For each convex set we can solve for

$$\min_{\underline{p}_i^j \in \mathcal{W}_i} \; ||\underline{p}_i^j - \underline{q}^j||, \tag{5.7}$$

where $\underline{p}_i^j$ is the closest point in the i^{th} set to the current solution $\underline{q}^j$. The points $\underline{p}_i^j$ will in general not be the same (for $u \neq v$, $\underline{p}_u^j \neq \underline{p}_v^j$).

Let us group all m sets together and denote this set $\mathcal{W}$. Now, we make an $n \times 2$

matrix $\boldsymbol{P}$ where the i^{th} row of $\boldsymbol{P}$ contains a point on the i^{th} set. Let $\boldsymbol{Q}$ be a second $n \times 2$ matrix with a point per set. Clearly a matrix $\boldsymbol{R} = \alpha\boldsymbol{Q} + (1-\alpha)\boldsymbol{P}$ ($\alpha \in [0,1]$) also has rows containing points the lie in the m sets.

Let the $n \times 2$ matrix $\boldsymbol{Q}^j$ have the current best guess $\underline{q}^j$ replicated across all rows. Denoting a least-squares projection as proj() we can write

$$\boldsymbol{P}^j = \text{proj}(\mathcal{W}, \boldsymbol{Q}^j). \tag{5.8}$$

The rows of $\boldsymbol{P}^j$ denote the closest points to $\underline{q}^j$. Of course to solve the problem we need to find a single point simultaneously closest to all sets. Let $\boldsymbol{U}$ denote the $n \times n$ matrix where every $\boldsymbol{U}_{a,b} = 1/n$. We can substitute each individual closest-point with the mean of the set by calculating

$$\boldsymbol{Q}^{j+1} = \boldsymbol{U}\boldsymbol{P}^j. \tag{5.9}$$

This equation finds the matrix $\boldsymbol{Q}^{j+1}$, where every row is the same and is closest in a least-squares sense to $\boldsymbol{P}^j$. In fact this operation is the projection of $\boldsymbol{Q}^j$ onto the set $\mathcal{U}$ of $n \times 2$ matrices where each row is the same (a convex linear subspace). We write this as

$$\boldsymbol{Q}^{j+1} = \text{proj}(\mathcal{U}, \boldsymbol{P}^j). \tag{5.10}$$

Both $\mathcal{W}$ and $\boldsymbol{Q}$ are fixed convex sets. $\boldsymbol{P}^j$ denotes the least-squares projection of $\boldsymbol{Q}^j$ onto the convex set $\mathcal{W}$ and $\boldsymbol{Q}^{j+1}$ denotes the least-squares projection of $\boldsymbol{P}^j$ onto $\mathcal{U}$. We show a point $\boldsymbol{Q}^0$ (arbitrarily chosen), which converges to a point closest to three convex sets in Figure 5.3.

Thus, equations 5.8 and 5.10 are Projections Onto Convex Sets (POCS). POCS (Stark et al., 1998) is a widely studied optimisation procedure which here has two particularly interesting properties. First, it always converges to a single intersection

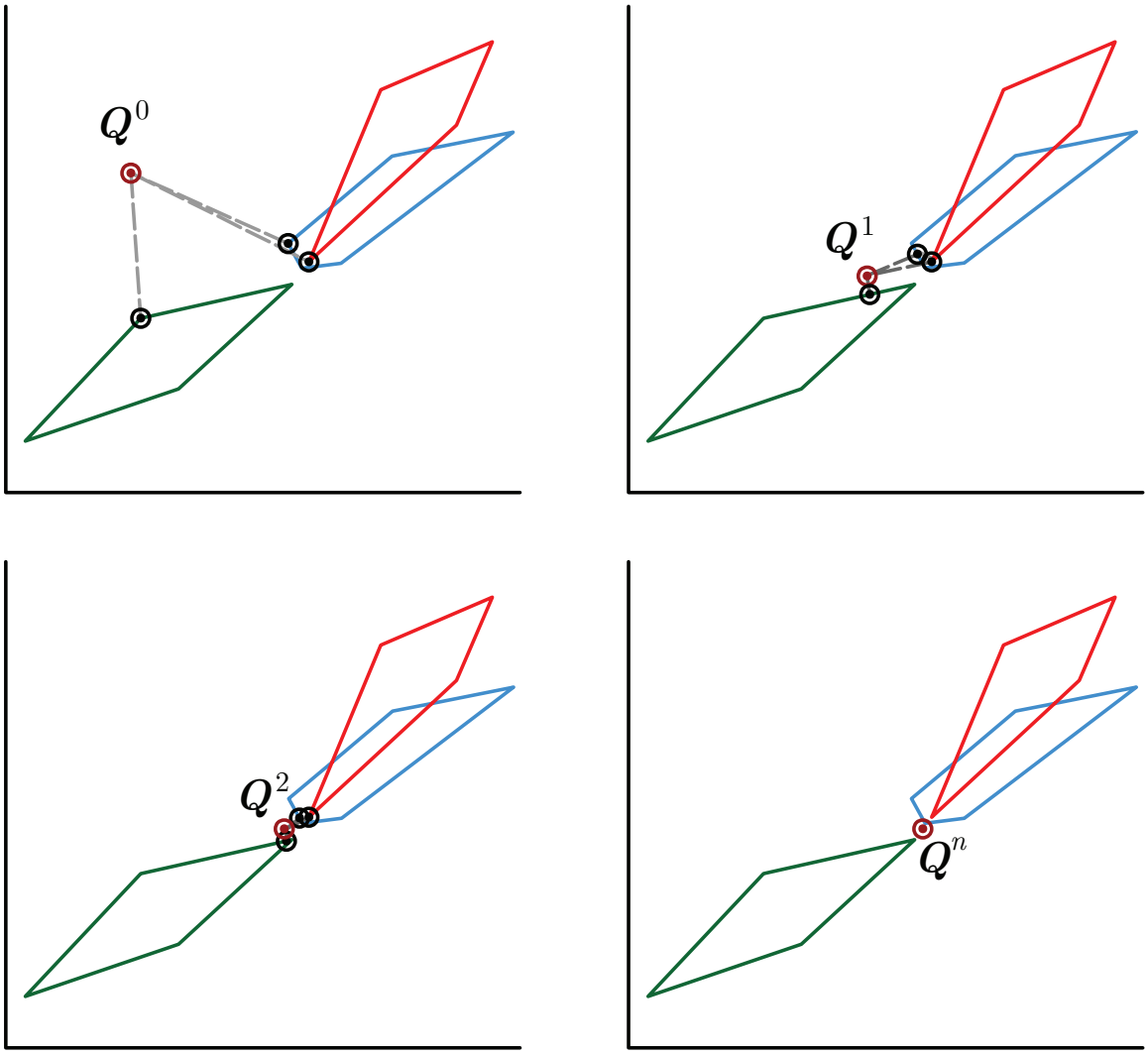

Figure 5.3: A visualisation of the convergence of a point $\boldsymbol{Q}^0$ to a least squares solution $\boldsymbol{Q}^n$ that is simultaneously closest to all sets.

(assuming the two sets have a non-null intersection). Because we have a set of n convex sets there can be at most 1 common intersection. Thus, in the case where all the line segments intersect, iterating between equations 5.8 and 5.10 will lead to the intersection point being found. In general however, there will be no common intersection point. For the special case of POCS operating on two non-intersecting convex sets then POCS returns a least-squares optimal solution. Our solution to the multiple lights colour constancy problem is summarised as:

1. $\boldsymbol{Q}^0$ (initialise in some reasonable way e.g. mean of centroids of all sets), $j = 0$.
2. $\boldsymbol{P}^j = \text{proj}(\mathcal{W}, \boldsymbol{Q}^j)$
3. $\boldsymbol{Q}^{j+1} = \text{proj}(\mathcal{U}, \boldsymbol{P}^j)$
4. *if* $||\boldsymbol{Q}^{j+1} - \boldsymbol{Q}^j|| < \delta$ stop *else* goto 2.

Where δ is a very small number.

5.3 Experiments

In this section we will experimentally evaluate our proposed algorithm. The nature of the algorithm means that it can manifest in several different forms. Traditional Gamut Mapping can be thought of as one manifestation. The line segment approach from Chapter 4 forms the foundation of another manifestation. Then there is also the unification of both methods. We predict that error in illuminant estimation will decrease the more information that is provided (i.e. the more lights and surfaces). In the first section we will do our experimentation on synthetic data. This allows us to generate a broad range of surfaces, all rendered under different illuminants. In the second section we once again use our rendering of the Gehler et al. (2008) dataset (described in Chapter 3). This is because each image contains a Macbeth Colour

Checker so the same 24 surfaces in 482 images taken on the same camera. We run iterations of the algorithm increasing the numbers of surfaces and lights used, and show the results in each case.

5.3.1 Synthetic Data

We once again synthetically generate pixels using the 1995 surface spectra and camera sensors provided by Barnard et al. (2002). For the illuminants we use the 99 daylights measured by Hernández-Andrés et al. (2001). Each surface is rendered under all of the illuminants, so we generate $1995 \times 99 = 197505$ pixels in total. For each pixel we can precisely calculate the correct illuminant RGB.

For the experimental design, we used 3-fold cross validation. For each fold we would generate surface and illuminant gamuts from two thirds of the of the respective data (i.e. two thirds of the surfaces rendered under two thirds of the illuminants). To generate the canonical surface gamut, RGBs were rendered as if viewed under a uniform white illuminant as $s_k = \sum_\omega S(\lambda)R_k(\lambda)$. We take the convex closure of the set to represent it in it's entitreity. For the illuminant gamut we generate illuminant RGBs as $e_k = \sum_\omega E(\lambda)R_k(\lambda)$, from two thirds of the illuminants. For testing, we used the remaining third of the surfaces rendered under the remaining third of the illuminants. The testing phase needs to represent different combinations of the algorithm. We evaluate combinations of increasing surfaces and illuminants. We test on all combinations of 1,2,4,8,16,32 surfaces under 1-4 illuminants. A 3-dimensional data store of 10,000 randomly chosen combinations of 32 surfaces under 4 lights, from the test set, was generated. We could then easily increase the number of surfaces and illuminants tested on for each variant of the algorithm by pulling from this data store. The only condition is that illuminants are more than 4 degrees in angular distance apart when we generate it. If two illuminants are too similar they

are not contributing any new information, so don't include sets that don't meet this criteria.

In Figure 5.4 we show the median result of this round of experiments. The median error for a single pixel (i.e. one surface under one light) is only 4°. It is low because of the illuminant and surface gamut constraints, which are able to strongly restrict the possible solutions given only a single pixel. In Figure 5.5 we have plotted the first two illuminants with 25-75 percentile error bars and in Figure 5.6 we have plotted the standard deviations. We see an increase in standard deviation when we use a lot of surfaces and a lot of illuminants. The error bars show us that this is because the 25^{th} has dropped with the 75^{th} percentile staying roughly the same. These are the cases where POCS is most likely to be used, as the more constraints the more chance of non-intersection.

The most profound change is between one and two illuminants, where we see median angular error drop by almost a quarter for a single surface. The performance is only matched by traditional gamut mapping when 16 surfaces are used. The mean result for two illuminants is still slightly higher than the median, implying some large errors. However the standard deviation is substantially lower for two illuminants.

To put this in context with our previous work we refer to Table 4.1 in Chapter 4 (page 79). For two illuminants on a single surface the line segment algorithm can produce a median angular error of 2.86° with a standard deviation of 3.02. This result is actually closer to what the algorithm described in this chapter produces with 4 lights and a single surface. We can explain this result by the weaker gamut constraint in this algorithm. By broadening the line segment to a gamut of illuminant mappings, we widen the possible solutions that the algorithm can output. For one surface - two lights, this algorithm does have a smaller standard deviation of 2.35. This implies less extremities in error. The line segment algorithm uses an optimisation step, which

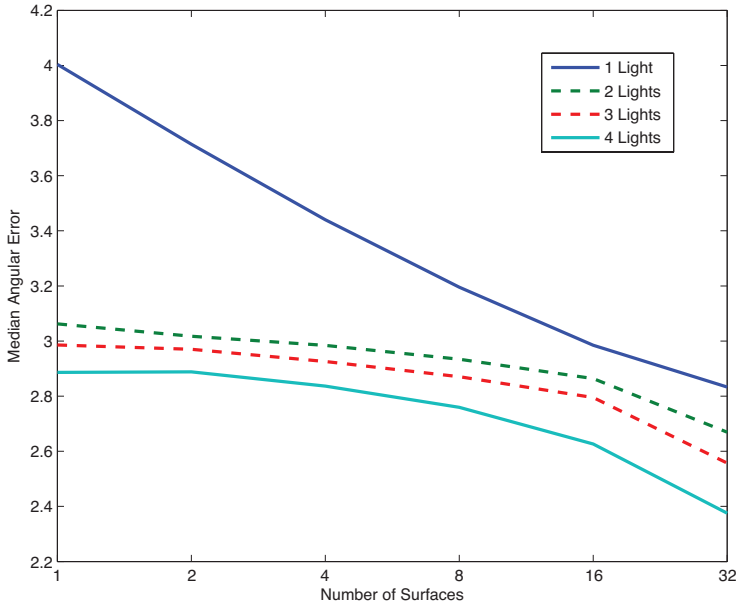

Figure 5.4: Median angular errors for varying surfaces and illuminants.

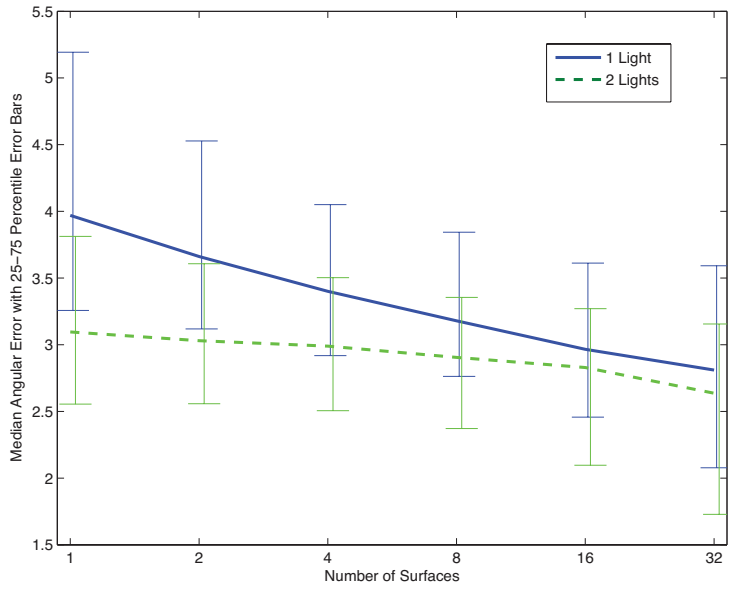

Figure 5.5: Median angular errors, with 25-75 percentile error bars for varying surfaces under the first 2 illuminants.

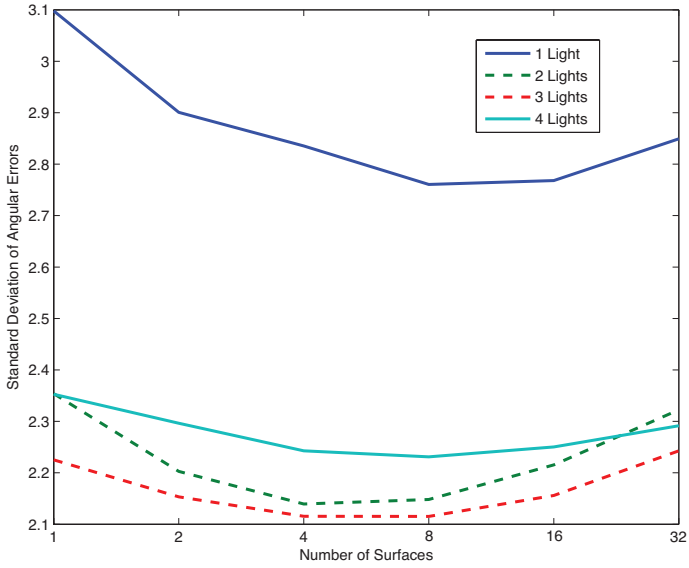

Figure 5.6: Standard deviations of angular errors for varying surfaces and illuminants.

tries to reduce the average error. The optimisation doesn't account for variance. By applying realistic gamut constraints, this technique appears to fail more gracefully.

This work does show that we can build and improve on gamut mapping, in scenes where there is more than one illuminant present (assuming we have successfully extracted the different illuminant data from the scene).

5.3.2 Macbeth Colour Patches from the Gehler et al. (2008) Dataset

To try and conceptualise performance on real data, we once again use our rendering of the Gehler et al. (2008) dataset (described in Chapter 3, page 53). We use 482 images taken on the Canon 5D SLR camera. In each image there are 24 patches from the Macbeth Colour Checker, so we have 24 surfaces that vary across different illuminants. We can use this captured data as input to the algorithm described in this chapter.

To assimilate the results from this section with the synthetic work of the previous section we would like to use the same canonical data. Because we are only using 24 surfaces in this section, which are well distributed across the colour space, doing the same cross-validation step in the previous section isn't appropriate. This is because any two-thirds subset is unlikely to be a representative canonical gamut. Instead, by using the reflectance spectra of a Macbeth Colour Checker we were able to synthetically generate 24 pixels of the colour checker in synthetic space. This meant we could map the 1995 synthetic RGBs from synthetic space to Gehler et al. (2008) camera space.

For the canonical illuminant set we did 3-fold cross validation on the 482 illuminants measured from the grey patches on the Macbeth Colour Charts. The illuminant is calculated as the mean of the non-clipped pixels in the grey patches. The algorithm is run for three folds where the surface canonical gamut is constant, but only colour charts from $482 \times 2/3 \approx 321$ are used to generate the illuminant gamut. The remaining colour charts are used for testing in one fold. This is cycled three times to cover all of the data. We show the Median results for 1,2,4,8,16 Macbeth Patches and 1-4 illuminants in Figure 5.7, along with the the 25-75 percentile error bars for 2 lights in Figure 5.8 and the standard deviations in Figure 5.9.

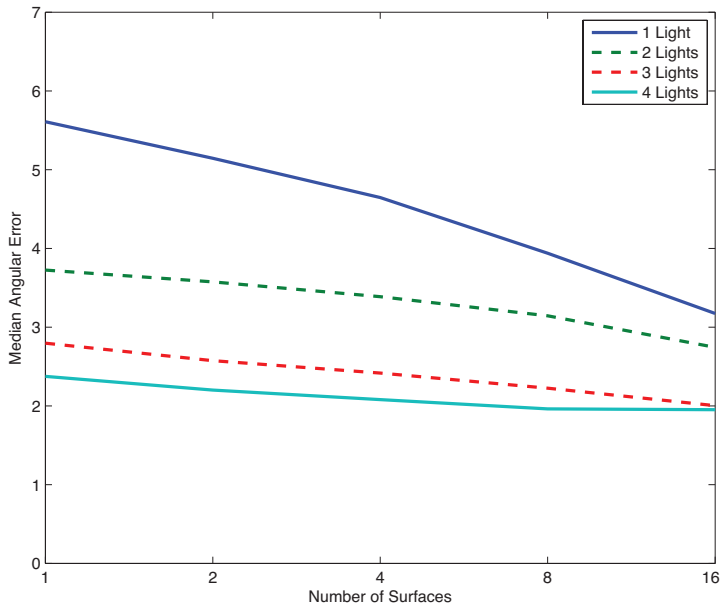

Figure 5.7: Median angular errors for varying surfaces and illuminants using Macbeth patches in the Gehler et al. (2008) dataset.

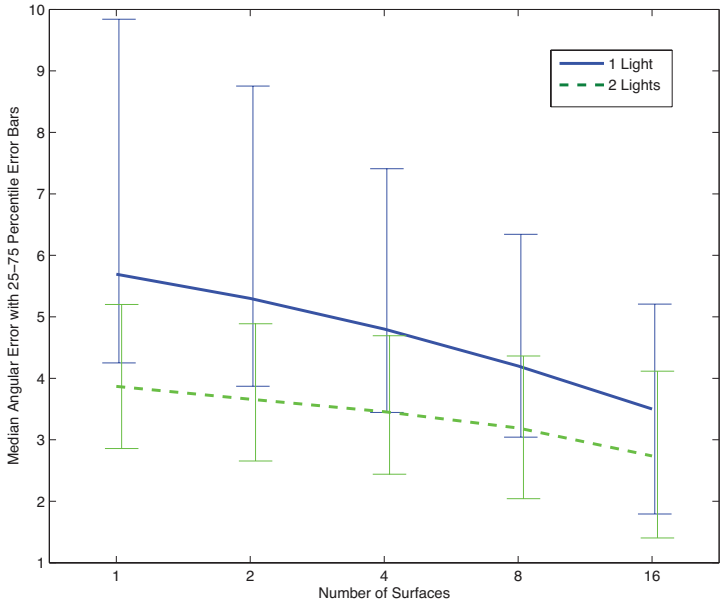

Figure 5.8: Median angular errors for 1-2 lights with 25-75 percentile error bars using Macbeth patches in the Gehler et al. (2008) dataset.

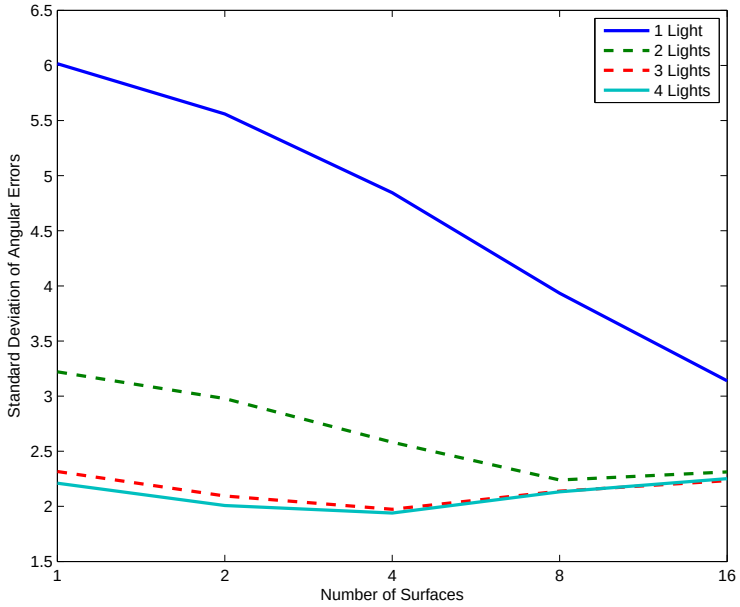

Figure 5.9: Standard Deviation of angular errors for varying surfaces and illuminants using Macbeth patches in the Gehler et al. (2008) dataset.

Immediately we see the same trend as the synthetic median data, with the biggest drop in a single step being from one to two illuminants (Figure 5.7). Gamut Mapping requires 16 surfaces to achieve a similar result to one surface two illuminants. We also know that these 16 surfaces are particularly well spanned, so in reality it may need more. We again see an increase in standard deviation when using more surfaces and illuminants (Figure 5.9). The error bars from Figure 5.8 show that more errors lie in the lower percentiles in this case.

We see a consistent improvement when increasing the number of illuminants, however it's affect gets less significant the more that are introduced. The comparatively modest difference between 16 surfaces one light (traditional gamut mapping) and 16 surfaces two lights highlights that the main factoring constraint is in one to two illuminants. This is unsurprising because of the predictability and natural constraints

on the illuminant set. Illuminant spectra behaves in a much more predictable way than surface reflectance spectra.

5.4 Conclusions

In this Chapter we have shown that we can broaden the applications of the Gamut Mapping framework by unifying it with our previous line segment work. Illuminants can be modelled as a convex set, and we have propagated this to a generalised algorithm which can be highly customised to the properties of an image. We have also described a methodology for dealing with the case of non-intersection of mapping sets. We provide a least-squares minimisation, which is in the format of a converging function. The function will converge to a point that is simultaneously closest to all sets.

Our Colour Constancy algorithm can estimate all illuminants in a scene, given segmented input. We acknowledge that segmenting areas of varying illumination in a scene is a hard problem to solve, which this work relies on. However, given that Gamut Mapping performance improves when framed in this way, the work provides a motivation for solving it.

The important result is that as the number of illuminants and surfaces increase, the average error decreases. Multiple illuminants in a scene often add many complications for illuminant estimation. When formulated in the right framework, the additional illuminant information is actually very helpful in terms of reducing the number of possible solutions. Traditionally Gamut Mapping algorithms have been applied to an entire image, with the assumption that there is a single scene illuminant to solve for. Clearly this is not the case - given that some scenes contain shadows which are essentially a change in illuminant colour.

There has shown to be a trade-off between using this algorithm and the line

segment algorithm described in Chapter 4, for 2 lights - 1 surface. By broadening the gamut of illuminants, we constrain answers to a realistic set of possibilities. When the line segment algorithm's assumptions do not hold, then the failure cases can be more extreme. In the next Chapter we explore a method to minimise the extreme errors but also keep the median low. We also look at the automatic detection of shadows in images, for a more complete illuminant estimation algorithm.

Chapter 6

Robust Two Illuminant Estimation with Shadow Edge Detection

6.1 Introduction

In this section we incorporate what we have learned from the previous chapters to form a complete colour constancy algorithm. The previous work requires the same surfaces under more than one light as input. The automatic collection of this data from images had not been addressed. Here we aim to automatically find pixels in an image that lie on a shadow edge. Specifically, we actually want pixels of the same surface on both sides of a shadow edge. This would provide data in the form that we need as input to the previous algorithms described in this thesis. One problem with the previous algorithms is that they require input data to be very accurate and aren't as robust to error. We attempt to address this here by formulating a colour constancy algorithm in a voting scheme. The assumption being that erroneous data would not be highly voted for.

6.1.1 Background

We have previously described algorithms which, given input data of the same surfaces under more than one light, return good estimates of the light colours. Up until now we have used hand-selected data for input, that we know conforms to the input

assumptions. For this work to have practical use we would like this process to be automated.

Barrow and Tenenbaum (1978) had the first recorded use of the term "intrinsic image". An intrinsic image, is a hypothetical image which contains only reflectance information. This means it would contain no 3D cues which come with illumination, such as gradients and shadows. Such an image would be useful in object recognition because it would be completely normalised against the effects of illumination colour so objects would appear the same under all lighting conditions. The recovery of the exact intrinsic image associated with a scene is impractical. However, removing shadows would be a step towards a rendering of the image under a single uniform illuminant.

If we aimed to automatically detect surfaces under more than one illuminant using only colour information (unlike the work of Panagopoulos et al. (2011), which requires higher level geometry information), we would have to expect a margin of error. This is because surfaces could plausibly have an illumination-like change at an edge. In this chapter we propose an illuminant estimation algorithm that is robust to such input errors. The robustness comes from an illumination voting scheme which weakly weights illuminant pair estimates that could be attributed to surface change.

The algorithms proposed in this thesis require surfaces under more than one light as input. This chapter provides a practical context for this requirement, by automating the detection of the same surfaces on different sides of a shadow edge. Consequently, varying illumination algorithms can be applied directly to an image without manual segmentation.

6.1.2 Colour-Based Shadow Recognition

One of the earliest algorithms which attempts to remove shadows from images was developed by Weiss (2001). The author's method involved capturing many images of scenes across different times of day. The position of the shadows in the scene would change as the sun moves across the sky. The authors propose taking the median over every pixel across all the images. They assume that each surface will be out of shadow most of the day, and so the median would represent the same light colour for all pixels. This effectively produces shadowless images most of the time. However, the method has no practical use when only one image is available, and requires almost a days worth of capture to get the required result.

Finlayson et al. (2002a) introduced a method for making greyscale images that are free from shadows. They examined the properties of Planckian Illumination. Planck's law is shown in the equation

$$E(\lambda_k, T) = Ia_1\lambda^{-5}\left(e^{\frac{a_2}{T\lambda_k}} - 1\right)^{-1}, \tag{6.1}$$

where a_1 and a_2 are equal to $3.74183 \times 10^{-16} Wm^2$ and 1.4388×10^{-2} respectively. The variable I represents the intensity of the illuminant and T represents colour temperature in Kelvin. For illuminants with a temperature range between 2500K and 10000K the term $e^{\frac{a_2}{T\lambda_k}}$ is $>= 1$ and so Wein's approximation (Wyszecki and Stiles, 1982)

$$E(\lambda_k, T) \approx Ia_1\lambda_k^{-5}e^{-\frac{a_2}{T\lambda_k}} \tag{6.2}$$

can be used. Both of these equations generate very smooth functions, which don't represent the wavelengths of actual illuminant spectra well (real spectra contains higher frequencies). They only begin to match because we integrate over all wavelengths

for RGBs. If we assume $R(\lambda)$ to be a perfect Dirac Delta function then we define $q_k = R(\lambda_k)$. Following this, we write the modified image formation equation as

$$p_k = S(\lambda_k)(Ia_1\lambda_k^{-5}e^{-\frac{a_2}{T\lambda_k}})q_k, \tag{6.3}$$

where $k = \{R, G, B\}$. Suppose we were to take the spectral band-ratio chromaticity of the pixels formulated by this equation. We have defined this previously as $[p_{R/B}, p_{G/B}]$. We remind the reader that this can be extended to dividing out any of the channels

$$c_k = \frac{p_k}{p_m}, \tag{6.4}$$

where $m \in \{R, G, B\}$. So $[p_{R/G}, p_{B/G}]$ and $[p_{G/R}, p_{B/R}]$ will also work. As a transition step we will now take the log of the chromaticity of a pixel (defined by equation 6.3). This will allow us to isolate the temperature term

$$\log(c_k) = \frac{\log(e_k - e_m)}{T} + \log\left(\frac{s_k}{s_m}\right), \tag{6.5}$$

where $s_k = a_1\lambda_k^{-5}S(\lambda_k)q_k$ and $e_k = -a_2\lambda_k$. As temperature changes then this equation forms a straight line in the 2-dimensional colour space. Being able to make images that are invariant to illumination colour change then involves calculating the direction of the 2D vector formed by $e_k - e_m$. A greyscale illumination invariant image $\mathcal{X}$ can therefore be defined as

$$\mathcal{X}^x = \underline{e}^{\perp}\log\left(\underline{c}\right), \tag{6.6}$$

where the two elements of $\underline{e}^{\perp}$ are constants that form the vector that is orthogonal to the direction of illumination change in log-chromatcity space. The authors have shown that shadows disappear in these greyscale images, demonstrated in Figure 6.1.

Figure 6.1: An image from the Gehler et al. (2008) dataset, with it's shadowless greyscale image, generated using Finlayson et al. (2002a)'s algorithm.

In later work, Drew et al. (2003) showed that you can retain some colour information from this process. As the line extends over log-chromaticity then the colour changes, this is not reflected in the greyscale image. To visualise the image we need replace the projection in the direction of $\underline{e}^{\perp}$ by a 2×2 projector $P_{\underline{e}^{\perp}}$. This forces the log-chromaticities to orient in the direction of $\underline{e}^{\perp}$ but preserves the 2D vector components. This is defined as

$$\mathcal{X} = P_{\underline{e}^{\perp}} \log(\underline{c}), \tag{6.7}$$

$$P_{\underline{e}^{\perp}} = \frac{\underline{e}^{\perp}(\underline{e}^{\perp})^T}{||\underline{e}^{\perp}||}. \tag{6.8}$$

We can then go to a standard chromaticity space for visualisation $p_k/(p_R + p_G + p_B)$. We display the same image as before, but retaining some colour information, in Figure 6.2. We can see in the image that shadows have been removed, retaining the redness of the walls and the greenness of the plant. This image behaves very similarly to an intrinsic image.

In later work Finlayson et al. (2009) set out to remove the calibration step involved in shadow removal. Currently, the direction of illumination change in log-chromaticity space is calculated for a specific capture device. Given that illumination change is defined as a direction, not a line, then we can represent it as an angle between 0° and 180°. The authors made the observation that by removing the shadows in the chromaticity image, then there is less colour variance. This means that the overall entropy of the image is lower. They argue that the angle that causes the image to have the smallest entropy will also be a shadow free image. This extension has made the shadow removal algorithm have application in images where the illumination direction in log-chromaticity space has not been calibrated for.

Figure 6.2: An image from the Gehler et al. (2008) dataset, with it's shadowless chromatcity image, generated using Drew et al. (2003)'s algorithm.

6.2 Finding Pixels on Both Sides of Shadow Edges

Here we wish to find pixel pairs where illumination changes but material properties are the same. We make the general assumption that there are two illuminants in the image, in-shadow and out-of-shadow. In Figure 6.2, illumination changes are absent, or at least greatly attenuated (such as in Figure 6.2). By comparing the gradient fields in the original image with those in the invariant image, we should be able to find edges that appear in the original but are not evident in the invariant.

The objective, however, is not to diminish these edges in the original, zeroing them out or inpainting across them (as is the work of Drew et al. (2003)), with a view to reintegrating the modified gradient field into a shadowless RGB image. Instead, we simply wish to identify illumination-change boundaries with the aim of establishing chromaticities on either side of the edge, for our illumination-voting scheme. We make use of a Mean-Shift segmentation (Comaniciu and Meer, 2002) of the image for establishing relevant edges. Mean-Shift is a common segmentation technique which replaces each pixel with the mean of the pixels in a neighbourhood. This has the advantage that shadow-edges can be quite blurry, in that what we are after is colour on either side: and the Mean-Shift segments can very simply supply an estimate of RGB triples on either side without need for pixel-level accuracy. This is because each segment draws upon many pixels to colour its region. This represents an advance over the necessity for sharp illumination edges (like in the work of Finlayson et al. (2002a)).

6.2.1 Detecting Shadow Edges

Suppose we have used Drew et al. (2003)'s invariant-image formulation to generate a chromaticity, illumination-free image. We shall denote this image by $\mathcal{X}$, a 3 channel image where each of the pixels sum to one. We form the Mean-Shift segmentation of

the input image; let us call this RGB segmentation image $\mathcal{R}_M$. An example Mean-Shift segmentation is shown in Figure 6.3. The nature of segmentation means that rigid edge boundaries are defined as part of the process. Colour is used directly to help us segment the input image. Rather than using binary edge maps, we are interested in gradients. We can use Canny (1986)'s edge finding mechanism; but here we use it to generate a gradient pair $\mathcal{R}_x$, $\mathcal{R}_y$ instead of a binary output, making use of a grayscale version $\mathcal{G}$ of $\mathcal{R}_M$ formed by summing the channels of $\mathcal{R}_M$.

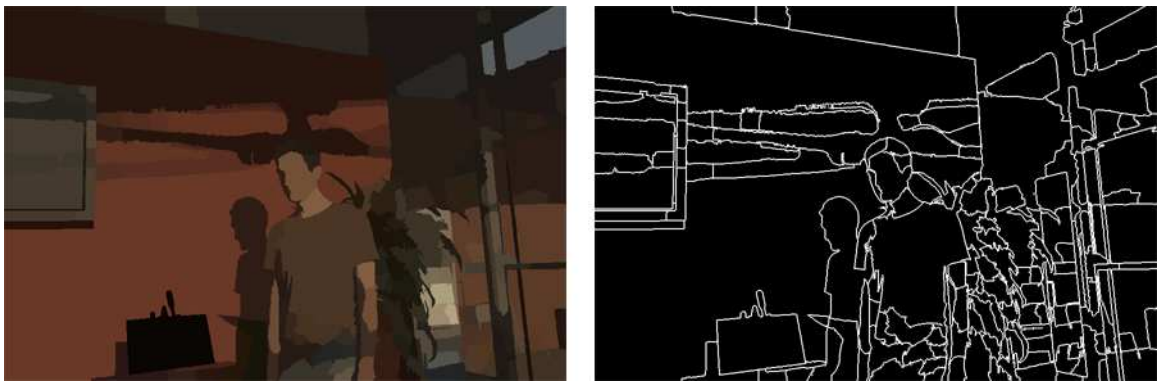

Figure 6.3: **Left:** Image split into segments using the Mean-Shift (Comaniciu and Meer, 2002) algorithm. **Right:** Rigid edges defined by Mean-Shift (binary output).

Similarly we calculate the Mean-Shift segmentation of the invariant $\mathcal{X}$. We refer to this 3 channel image as $\mathcal{X}_M$. Figure 6.4 shows an example of this process. We can see that segments which were attributed to illumination change have been removed. To make a simple gradient field out of this chromaticity image, we would once again like to form a grey image G as the sum of colour bands. However, chromaticities add to unity; therefore instead we use a grey formed in the L2 norm, as $G = (\sum_{k=1}^{3} \mathcal{X}_k^2)^{1/2}$. Denote the gradient pair thus formed out of the invariant-chromaticity image by G_x, G_y.

We are now interested in discovering the edges solely accounting for illumination change. In terms of the images we have generated so far, we are interested in edges

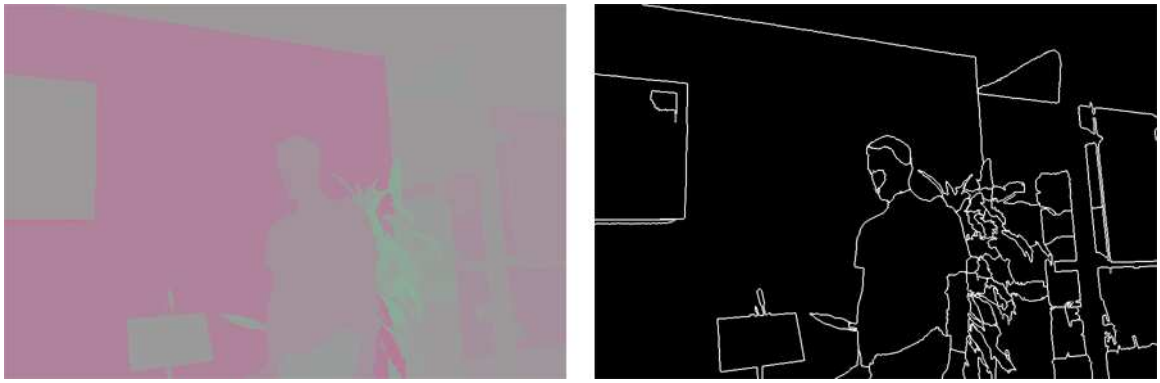

Figure 6.4: **Left:** Invariant chromaticity image split into segments using the Mean-Shift (Comaniciu and Meer, 2002) algorithm. **Right:** Rigid edges defined by Mean-Shift.

that do appear in the original but not in the invariant. Given our two gradient fields $\mathcal{R}_x$, $\mathcal{R}_y$ and G_x, G_y we perform non-maximum suppression as in Canny edge detection, using 4 quadrants of directions. The 4 quadrants beings top-bottom, left-right and the the opposing diagonals. Because of the pixel grid we cannot sample any finer than that. Non-shadow edges are removed by removing edges occurring in the original image with the same orientations as edges occurring in the same location in the invariant image. Finally, noting that isolated dots and lines in the original image segmentation appear weakened in the invariant-chromaticity image, these pixels are removed from the generated shadow-edge map. The result is a set of 1 pixel-wide shadow boundaries, bordering on input-image Mean-Shift edges (as shown in Figure 6.5).

6.2.2 Culling Edge Pixels

For this application it is not required to be completely successful in identifying all shadow-edge pixels - see, e.g., Figure 6.5, showing candidate edges identified. As well, we do not need every instance of the same across-edge RGB pairs from the Mean-Shift

Figure 6.5: Image with Mean-Shift Shadow-Edge estimates.

image of the input RGB (even a single pixel pair would do, in theory).

We next find across-edge pairs by finding, for each pixel on a Mean-Shift image edge, the closest pixel in a different segment than the pixel's own segment. This always works, to find across-edge colour pairs (and is fast). We then have one RGB triple and a second RGB triple for each illumination-edge pixel. We put these in luminance order by asking which triple is darker (has smaller R+G+B), and then remove all but the unique 6-vector (3 plus 3) RGB pairs, RGB1 and RGB2 . At this point we can remove further edge pixels by asking, amongst all the pixel-pairs, which ones are well described by a single representative 3×3 transform M from RGB1 to RGB2. To do so, we make use of the Least Median of Squares (LMS) regression (Rousseeuw, 1984), which automatically identifies outliers. The LMS procedure also delivers a robust

version of the Coefficient of Determination, which must be sufficiently high in order to endorse having indeed found illumination boundaries, in the current input image.

At this point, we have a much reduced set of RGB-triple pairs, for which we can apply our illuminant-voting algorithm.

6.3 Voting Algorithm

In this section we describe a new two-illuminant colour constancy algorithm, which includes an extended model of image formation. The algorithm is formulated as a voting strategy, in which a set of different illuminants are voted for based on the input data. We provide context for the application of this algorithm, as we have already described a method for the automatic-detection of pixels on either side of a shadow edge. The output from this process is used directly as input to our voting algorithm, which then outputs an estimate for each of the two illuminants.

6.3.1 Two-Illuminant Estimation by Voting

A camera system is unlikely to have completely narrowband sensors, but they are often narrowband enough for the diagonal model assumption to approximately hold (Finlayson et al., 1994). However, the more saturated a pixel, the more error that will be introduced into the diagonal model. Previously we have modelled image formation at a pixel as $\underline{p}^x \approx \boldsymbol{E}\underline{S}$, where the diagonal elements of $\boldsymbol{E}$ are the RGBs of the illuminant. Consequently we could turn $\boldsymbol{E}$ into a mapping matrix $\boldsymbol{E}^{-1}$ by just taking the inverse of it's diagonal elements. We can improve accuracy of this mapping model by incorporating all 9 elements into of a 3×3 mapping matrix such that

$$\begin{bmatrix} S_r \\ S_g \\ S_b \end{bmatrix} \approx \begin{bmatrix} M_{1,1} & M_{1,2} & M_{1,3} \\ M_{2,1} & M_{2,2} & M_{2,3} \\ M_{3,1} & M_{3,2} & M_{3,3} \end{bmatrix} \begin{bmatrix} p_r^x \\ p_g^x \\ p_b^x \end{bmatrix}, \tag{6.9}$$

where S_k are the RGB elements of the surface reflectance and p_k^x are the RGB elements of the pixel response. By introducing more degrees of freedom, the model has more accuracy. However this model causes extra complexity, for example: it does not translate directly to chromaticity space. We would like to incorporate this model into our algorithm to allow for more accurate illuminant mappings. Suppose we capture a collection of surfaces under a representative set of real-world illuminants. For example, we could capture the 24 patches of a Macbeth ColourChecker at different times of day throughout the year. Assuming each patch was captured uniformly under the same illuminant then we have a single illuminant RGB $\underline{E}^\epsilon$ for each capture of the surface set. We can formulate our set of n surfaces as an $n \times 3$ matrix $\mathcal{S}^\epsilon$. Suppose we also have the surface set under some canonical illuminant $\underline{E}^c$, such as a uniform white light or D65, we can call this $\mathcal{S}^c$. We can solve for a 3×3 illuminant mapping matrix $\boldsymbol{M}^\epsilon$ using least-squares to get

$$\boldsymbol{M}^\epsilon = ([\mathcal{S}^\epsilon]^T \mathcal{S}^\epsilon)^{-1}([\mathcal{S}^\epsilon]^T \mathcal{S}^c), \tag{6.10}$$

where T represents the matrix transpose. So for each illuminant $\underline{E}^\epsilon$ we can associate a 3×3 illuminant mapping matrix $\boldsymbol{M}^\epsilon$. Suppose we have two pixels of the same surface captured under two lights $\underline{p}^1$ and $\underline{p}^2$. Then $\alpha \boldsymbol{M}^1 \underline{p}^1 = \beta \boldsymbol{M}^2 \underline{p}^2 = \underline{p}^c$. Where $\underline{p}^c$ is the pixel response captured under the canonical light c, and α and β are arbitrary scalers (we do not expect to recover brightness). Now suppose we had captured a surface under two illuminants, but did not know what the two illuminants were. If we were to choose some estimate arbitrarily for both illuminants $\boldsymbol{M}^a$ and $\boldsymbol{M}^b$, we can evaluate their plausibility by measuring how similar $\boldsymbol{M}^a \underline{p}^1$ is to $\boldsymbol{M}^b \underline{p}^2$. As we expect brightness to vary, then we propose using an intensity-invariant measure. For a given $\underline{p}^1$ and $\underline{p}^2$ we use angular error as a measure of likelihood that a and b are the correct illuminants, so

$$\theta^{a,b} = \text{ang}\left(\boldsymbol{M}^a \underline{p}^1, \boldsymbol{M}^b \underline{p}^2\right). \tag{6.11}$$

If $\theta^{a,b}$ is large, it means that illuminants a and b are poor estimates for the illuminant. Given some representative set of n illuminants, we could use this model to test the likelihood of any pair being the correct pair of illuminants. If we test all the pairs of n illuminants for a single surface then we would calculate $n^2 - n$ angles (we minus n because we don't calculate angles for the same illuminant). We could now just pick the illuminant pair with the smallest angle. However, in a scene we may have pairs of many different surfaces under the same two lights (i.e. along a shadow edge). We would like to frame the algorithm in a way that makes use of this information. We convert the $n^2 - n$ angles into meaningful votes for each light. First, we map our angular measure to the interval [0,1] so

$$v(\theta, \sigma) = \exp\left(\frac{-\theta^2}{2\sigma^2}\right). \tag{6.12}$$

The smaller σ the more bias the voting scheme will have towards smaller angles (small angle = big vote). We can extend this model to allow surfaces to vary, given two illuminants under a surface x, we can calculate a vote as $v(\theta^x_{a,b}, \sigma)$. We would like to run two voting polls, one for each illuminant. A voting poll is ran for each of the n illuminants in our representative set. Suppose we have pairs of m surfaces under two lights as input, then we define two voting polls as

$$\text{poll } 1[a] = \sum_{x=1}^{m}\sum_{b=1}^{n} v(\theta^x_{a,b}, \sigma), \tag{6.13}$$

$$\text{poll } 2[a] = \sum_{x=1}^{m}\sum_{b=1}^{n} v(\theta^x_{b,a}, \sigma), \tag{6.14}$$

where poll $1[a]$ and poll $2[a]$ are votes for the a^{th} illuminant being correct for it's respective pixel. We do this for all n illuminants. A graphical example of a reddish

and bluish illuminant being voted for in two polls is shown in Figure 6.6.

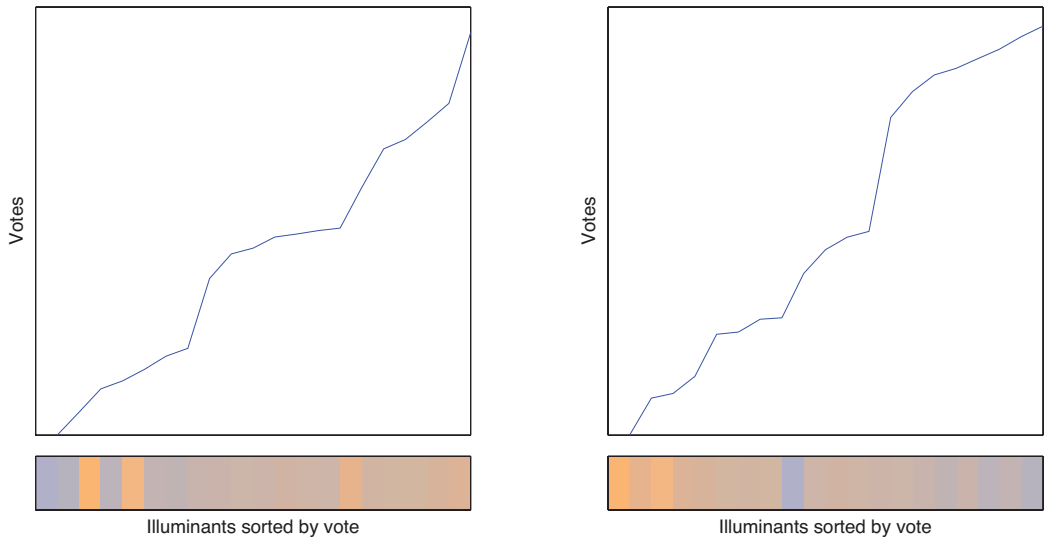

Figure 6.6: An example pair of illuminant voting polls, sorted in ascending order of votes.

Given these two voting polls the final step is to choose a single estimate for both illuminants. We could just choose the illuminant with the highest vote. However, suppose several of our test illuminants circle the correct answer, which isn't in our set: then they will yield an equally high vote. Taking the average of these will produce a better result than just picking one. We propose thresholding illuminants that are highly voted for and use their vote as a weight to output a weighted average. This has shown to perform better than choosing the pair of illuminants with the smallest angle. We have found that weighting the lights that account for the top 20% of votes yielded good performance.

6.3.2 Generating the Full Illuminant Set

Ideally we would like to choose an illuminant set such that they uniformly sample the illumination space. Given a set of representative illuminants, we can take their

convex closure in chromaticity space (like in the work by Finlayson (1996)). This is because we assume that any convex combination of the illuminants in our set is also a plausible illuminant. This is the full set of illuminants that we would like to represent. If we randomly sample points in RGB space, we can project those points into 2D chromaticity space. We should only keep the points that lie within the convex set. We now have a representative set of illuminant chromaticities. However, we require a 3×3 mapping matrix $\boldsymbol{M}^{\epsilon}$ to be associated with each illuminant $\underline{E}^{\epsilon}$. Here we look at solving for the 3×3 mapping matrix for illuminant RGBs that were not explicitly captured. Given a set of n representative illuminants we can define a $n \times 9$ polynomial expansion matrix $\mathcal{E}$

$$\mathcal{E} = \begin{bmatrix} E_r^1 & E_g^1 & E_b^1 & E_r^1 E_r^1 & E_g^1 E_g^1 & E_b^1 E_b^1 & E_r^1 E_g^1 & E_g^1 E_b^1 & E_r^1 E_b^1 \\ E_r^2 & E_g^2 & E_b^2 & E_r^2 E_r^2 & E_g^2 E_g^2 & E_b^2 E_b^2 & E_r^2 E_g^2 & E_g^2 E_b^2 & E_r^2 E_b^2 \\ .. & .. & .. & .. & .. & .. & .. & .. & .. \\ E_r^n & E_g^n & E_b^n & E_r^n E_r^n & E_g^n E_g^n & E_b^n E_b^n & E_r^n E_g^n & E_g^n E_b^n & E_r^n E_b^n \end{bmatrix}. \tag{6.15}$$

In our set of known illuminants, we know their associated 3×3 mapping matrix $\boldsymbol{M}^{\epsilon}$. We also define the following $n \times 9$ matrix $\mathcal{M}$, where the rows are the 3×3 illuminant mappings

$$\mathcal{M} = \begin{bmatrix} M_{1,1}^1 & M_{1,2}^1 & M_{1,3}^1 & M_{2,1}^1 & M_{2,2}^1 & M_{2,3}^1 & M_{3,1}^1 & M_{3,2}^1 & M_{3,3}^1 \\ M_{1,1}^2 & M_{1,2}^2 & M_{1,3}^2 & M_{2,1}^2 & M_{2,2}^2 & M_{2,3}^2 & M_{3,1}^2 & M_{3,2}^2 & M_{3,3}^2 \\ .. & .. & .. & .. & .. & .. & .. & .. & .. \\ M_{1,1}^n & M_{1,2}^n & M_{1,3}^n & M_{2,1}^n & M_{2,2}^n & M_{2,3}^n & M_{3,1}^n & M_{3,2}^n & M_{3,3}^n \end{bmatrix}. \tag{6.16}$$

Using least squares we can now solve for the 9×9 matrix $\mathcal{H}$ which associates illuminant RGBs with 3×3 mapping matrices

$$\mathcal{H} = (\mathcal{E}^T \mathcal{E})^{-1} (\mathcal{E}^T \mathcal{M}). \tag{6.17}$$

Given this 9×9 matrix $\mathcal{H}$, we can map illuminant RGBs to 3×3 illuminant mappings. We have found that this relationship can be well modelled by a polynomial expansion for real camera sensors.

6.4 Experiments

In this section we will put our algorithm in context with other algorithms. We initially test the performance of the voting algorithm in isolation on synthetic and real data. For this we provide the same same surface under two illuminants. We then examine the performance of the algorithm using the input from our shadow-edge finder, by running it on a common colour constancy dataset, as well as our own shadow datasets.

6.4.1 Experimental Design

In our scenes we assume the presence of two dominating light conditions. To have a ground-truth value for the chromaticity of these lights, the patches of a Macbeth Colour Chart were used. The white points were calculated as the mean average of the non-clipped grey patches. To generate our illuminant set we use the methodology described in Chapter 3, Section 6.3.2; Page 120. Because our algorithm requires a calibration step, it would be bias to run it on the same data it was trained on. For our experiments we ran a 3-fold cross validation on the algorithms that required calibration.

6.4.2 Synthetic Data

Here we will assess the performance of the voting algorithm on pixels synthetically generated using measured spectral data. We use the 1995 measured surface reflectances and the Sony DXC-930 camera sensitivities compiled by Barnard et al. (2002) and the 99 daylight spectra measured by Hernández-Andrés et al. (2001). We

can synthetically generate pixels according to the Lambertian model of image formation. In this section we will examine the case of a single surface under two lights. A surface is chosen at random and then rendered under two different randomly chosen illuminants. Some of the illuminants in the Hernández-Andrés et al. (2001) set are almost metameric, therefore we choose pairs of illuminants such that the rendered pixels are more than 4 degrees in angular distance apart. This is a realistic constraint because shadows can easily generate a colour difference of this or more. We can also assess the pixel responses that we are running the algorithm on are sufficiently different in advance. Since we have access to the surface reflectance spectra of the Macbeth Colour Chart, we can use this to calculate the 3×3 mappings for each illuminant. We generate pairs of pixels this way, uniquely, 5,000 times. We use 3-fold cross validation so the experiment is run a total of 15,000 times to incorporate the entire dataset.

For comparison we also include the result from Chapter 4, the line segment 1 surface 2 light algorithm. We show the mean and median angular errors of all the algorithms as well as the standard deviation in Table 6.1.

Algorithm	Median	Mean	Standard Deviation
Finlayson et al. (1995)	22.20°	36.64°	35.21
Kawakami and Ikeuchi (2009)	8.90°	11.93°	55.79
Kawakami and Ikeuchi (2009) (Modified)	3.46°	5.04°	4.28
Line Segment (Chapter 4)	2.86°	3.74°	3.02
Unifying Gamuts (Chapter 5)	3.08°	3.21°	2.36
Voting Algorithm	2.81°	3.70°	3.08

Table 6.1: Table comparing the angular errors of 2 illuminant 1 surface algorithms to our voting algorithm on synthetic data.

Our line segment algorithm has shown to perform well in this circumstance. We show here that we can achieve similarly good performance with the voting algorithm. The unifying gamuts approach has a higher median but a lower mean and standard

deviation. This shows that the gamut constraint is better at reducing the higher percentiles of error.

Next we want to benchmark our voting algorithm against some other standard colour constancy algorithms. These include the Gijsenij et al. (2011) implementation of Gamut Mapping. The code requires a minimum of 5 surfaces as input, so 5 surfaces under the same light are chosen at random. Given that the Gamut Mapping algorithm's performance is known to increase with the number of surfaces, this will work in it's favour. As there is a calibration step in calculating the canonical gamut, we once again do 3-fold cross validation. For comparison we run our voting algorithm with 5 surfaces under 2 lights. This is ran for 5000 iterations per fold, giving 15,000 iterations in total. Gamut mapping is run on the 5 surfaces under the two lights separately to gain the two illuminant estimates. The same is done for the GreyWorld and MaxRGB algorithms. The results are shown in Table 6.2.

Algorithm	Median	Mean	Standard Deviation
Gamut Mapping (Forsyth, 1990)	10.23°	11.66°	11.68
GreyWorld (Buchsbaum, 1980)	8.56°	9.69°	5.97
MaxRGB (Land and McCann, 1971)	9.31°	10.85°	7.64
Voting Algorithm	2.77°	3.57°	3.04

Table 6.2: Table comparing transitional Colour Constancy algorithms to our voting algorithm using synthetic data.

The results in this table clearly show that when there is little surface information, having access to even a single surface under 2 lights is a very powerful constraint. Gamut mapping, known for it's good performance, performs less well when it has a small amount of input surfaces. It's median performance being worse than the much simpler algorithms GreyWorld and MaxRGB. Gamut mapping's performance, when an entire scene worth of surface information is available, is known to perform much better. However this presents an interesting result, which emphasises the usefulness

of the light constraint.

6.4.3 Experiments on the Colour Chart Set

In this section we use our rendering of 482 images from the Gehler et al. (2008) dataset, discussed in Chapter 3. Since there is a colour chart in every image containing the same 24 surfaces, then we have 24 surfaces under multiple lights to test our algorithm on. We, once again, run experiments on our voting algorithm using 3-fold cross validation. The illuminant is calculated as the mean of the non-clipped grey patches on the colour chart. We run Gamut Mapping on 5 randomly chosen surfaces for comparison. Here we shall use the convex hull of the 24 patches to generate the canonical gamut. This is actually unfairly biasing the algorithm, because the canonical gamut is a known representation of the input data. We do this to show it performing at it's best. For similarity we also run MaxRGB and GreyWorld on the same 5 surfaces. Finally we run the single surface two light line segment algorithm from Chapter 4. The results are shown in Table 6.3.

Algorithm	Median	Mean	Standard Deviation
Gamut Mapping (Forsyth, 1990)	2.32°	3.01°	3.97
GreyWorld (Buchsbaum, 1980)	6.31°	7.34°	4.62
MaxRGB (Land and McCann, 1971)	5.46°	6.52°	6.48
Line Segment (Chapter 4)	4.31°	6.29°	5.08
Unifying Gamuts (Chapter 5)	2.19°	2.78°	3.20
Voting Algorithm	2.12°	3.91°	4.02

Table 6.3: Median, Mean and Standard Deviation of Angular Errors for different algorithms run on 5 surfaces on Macbeth Colour Checker chosen at random. The surfaces are taken from our rendering of the Gehler et al. (2008) dataset. We compare these to the result of our Voting Algorithm as well as the 2 algorithms outlined in previous chapters. These are ran on a single surface under two lights.

On real data our results emulate the synthetic data. As previously shown information about the same surface under two illuminants is a much more powerful constraint

than many surfaces under a single illuminant. Gamut mapping performs much better here because the canonical gamut is so representative of the input data. Also, since the surfaces on the Macbeth Colour Chart are broadly coloured, then choosing 5 at random is likely to make a large gamut. This means that the output mapping set will be small. As a result the Unifying Gamuts approach also performs very well, with only a slight improvement when introducing the additional light information. Even with all of the variables stacked in the favour of this algorithm, our voting algorithm has the best median performance. Once again, Unifying Gamuts has the smallest mean and standard deviation.

Experiments from Selected Images

In this section we examine the combined use of the shadow-edge detector and the voting algorithm. Most images in the Gehler dataset were taken on overcast days, which meant the ambient lighting was mostly uniform. However, we have chosen a subset of 52 images which do contain notable shadows. For each shadow-edge pixel found, the shadow-edge detector outputs two pixels for the same surface under two illuminants. We directly feed this into the voting algorithm to get an estimate for the illuminant colour on both sides of the shadow edge. We are only able to measure the illumination colour on one side of the shadow edge because each image contains only a single Colour Chart. Our error measures will therefore take into account only the illuminant that the Colour Chart is under. For this round of experiments, 3-fold cross validation was not necessary. This is because we can use the 430 unused images from the dataset to build our set of illuminant maps. We once again benchmark against the Gamut Mapping algorithm, and use the 430 unused images to build the canonical gamut. The pixels in the images were divided by their white patch to build the gamut. The results for this round of experiments are shown in table 6.5.

Algorithm	Median	Mean	Standard Deviation
Gamut Mapping (Forsyth, 1990)	3.22°	5.67°	5.92
GreyWorld (Buchsbaum, 1980)	4.37°	5.50°	6.13
MaxRGB (Land and McCann, 1971)	4.38°	5.34°	6.11
Grey Edge (Weijer and Gevers, 2007)	2.62°	4.48°	4.18
Unifying Gamuts (Chapter 5)	3.12°	3.71°	3.98
Voting Algorithm	2.10°	4.24°	4.02

Table 6.4: Angular Errors of different algorithms using 52 selected images from the Gehler et al. (2008) dataset which contain shadows.

The voting algorithms median error is significantly lower than the other algorithms. Gamut Mapping in this case is not the best performing of the other algorithms. This is unsurprising because it is expecting surfaces under a single illuminant colour as input to build the image gamut. The higher mean of our voting algorithm indicates some poor performance on some images. However, it is still lower than the other algorithm's mean result. The lowest mean is from the Unifying Gamuts approach, which shows a slight increase in performance than just using Gamut Mapping, particularly in the more extreme cases with the big drop in the mean result.

6.5 Custom Dataset

To further evaluate our method we tested on the two new datasets captured using the Nikon D60 SLR camera from Chapter 3. The first being a particularly hard dataset for standard colour constancy algorithms because it contains very limited surface variation and two dominant illuminants. The second is a more realistic datasets of different scenes containing shadows. A colour chart was placed inside and outside of shadow edge so that the two white points could be extracted.

6.5.1 Colour Patch Shadow Dataset

For the first dataset, 58 images were taken using matte colour patches (Chapter 3, Page 59). Each image contains a different combination of 5 patches laid over a shadow edge. The images were taken at different times of day and at different locations in the UK. In some of the images artificial lighting was used to make a shadow edge. A Macbeth Colour Chart was placed on both sides of the shadow edge, so that the 2 correct illuminant colours could be calculated. The mean of the unclipped grey patches was used as the illuminant colour. This dataset presents a particularly hard case for standard Colour Constancy algorithms, because the colour variation is small, and there are two lights in the scene.

We first manually crop out the 5 patches under the 2 illuminants in each image. This equates to 10 RGBs per image. We can input these RGBs directly into our voting algorithm to get an estimate for the 2 illuminant RGBs. We use 3-fold cross validation to train and test on this dataset. For comparison we also compare our result to our Line Segment algorithm from Chapter 4 and our Unifying Gamuts algorithm from Chapter 5. We also compare against some traditional one illuminant algorithms, by inputting the surfaces that are in and out of shadow separately. The results are shown in Table 6.5.

Algorithm	Median	Mean	Standard Deviation
Gamut Mapping (Forsyth, 1990)	3.19°	5.61°	4.02
GreyWorld (Buchsbaum, 1980)	13.81°	15.12°	9.06
MaxRGB (Land and McCann, 1971)	16.12°	16.22°	10.55
Line Segment (Chapter 4)	3.41°	4.61°	3.60
Unifying Gamuts (Chapter 5)	2.87°	3.94°	3.43
Voting Algorithm	2.73°	4.01°	3.56

Table 6.5: Angular Errors of algorithms run on Syntha Pulvin patches in and out of shadow.

This dataset was designed to be difficult for traditional colour constancy algorithms, so it is unsurprising that we see poor performance for GreyWorld and MaxRGB. Gamut Mapping does much better in this case, this is probably due to the representative canonical gamut used (Chapter 3; Figure 3.11, page 61). We see particularly good performance with the two illuminant algorithms, compared to the traditional one illuminant algorithms. We also see the same pattern occurring with the Voting Algorithm outperforming the others in terms of median but the Unifying Gamuts approach having the lower mean and standard deviation.

6.5.2 Shadow Images

In this section we use the images from Chapter 3; Section 3.4.2 (page 65). They are real-world scenes captured at different locations and at different times of day. All of the images contain shadows, and thus have at least two illuminant colours. To record the illuminant colours a Macbeth Colour Checker was placed in and outside of shadow. Using these images we will apply our complete illuminant estimation algorithm, which incorporates the automatic detection of shadows edges. In Figure 6.7 we show an example image from the dataset with shadow edges found. There are some errors in estimation, which is to be expected since the algorithm uses only the colour information to detect edges. There are surfaces changes which could be plausibly mistaken for an illumination change. However, this is minimised by using the Robust Median Regression, to find mappings which best fit to 50% of the data. The nature of the voting algorithm also adds a level of robustness as pixel pairs of the same surface under the two scene illuminants should consistently vote for their respective illuminants. This should outweigh the effect of a surface edge having a shift colour that is similar to an illuminant shift. This is because the effect is unlikely to repeat in the same way across the scene in the way actual illuminant change does.

Figure 6.7: **Top Left:** Shadow Edges Detected **Top Right:** Mean Shift Image **Bottom Left:** Shadowless Greyscale Image **Bottom Right:** Shadowless Chromaticity Image

The shadow edge estimation step outputs pairs of pixels which are considered to be the same surface but under two lights. We can use this as direct input to our voting algorithm estimator, and make two predictions per image for each illuminant. Since we have measured the correct illuminants then we can take the angular error for each estimation. For comparison we also, once again, compare the results to three traditional colour constancy algorithms (MaxRGB, GreyWorld, Gamut Mapping). We use only the surfaces taken from each side of the shadow edge to make an estimation. For Gamut Mapping, we generate the canonical set by using the selection of coloured patches under a single illuminant from the Syntha Pulvin dataset. We can do this because both datasets were captured on the same camera. We also compare

the result with our work from Chapter 5, which unifies surface gamut constraints with multiple illuminant constraints. We present the results of this step in Table 6.6.

Algorithm	Mean	Median	Standard Deviation
Gamut Mapping (Forsyth, 1990)	5.23°	5.99°	4.13
GreyWorld (Buchsbaum, 1980)	5.08°	5.22°	3.91
MaxRGB (Land and McCann, 1971)	6.48°	6.18°	3.35
Unifying Gamuts (Chapter 5)	3.03°	3.44°	3.26
Voting Algorithm	2.58°	3.51°	3.48

Table 6.6: New Nikon dataset, automatic shadow-edge plus voting

We see that traditional algorithm's do not perform as well as the multiple illuminant algorithms for this case. The approach from Chapter 5 improves the performance of Gamut Mapping by using the cross-correlated information from the surfaces under two illuminants. We once again see the trend of the voting algorithm having the lower median, but the Unifying Gamuts approach has a lower mean and standard deviation. The approach described in this Chapter is able to achieve the lowest error 50% of the time. The lower mean average for the Unifying Gamuts approach indicates that using Gamut Constraints are good for overall reduction of error (showing the same pattern that has emerged before).

6.6 Conclusions

We have proposed an algorithm which detects pixels on both sides of shadow edges in an image, and then uses this information to estimate the two illuminant colours. Further, we developed a voting strategy for illuminant estimation which we show provides a significant improvement over the prior art.

The results we present here are important for three reasons. First, images with two lights are notoriously hard (impossible) to solve for the majority of algorithms which assume there is a single illuminant present. Second, we show that we can find

illumination change in images using quite a simple algorithm. Third, even when there is just a single surface under two different lights, we get a very strong cue to what the illuminant colour actually is.

This work captures the zeitgeist of current illuminant estimation research in that we are not proposing to solve the whole problem but, rather, have identified a particular case which we can solve (and for which traditional algorithms fail). This leads us to believe that general algorithms for illuminant estimation should be engineered from more specialised components that are designed to solve particular classes of images (e.g. those with faces or those with shadow edges).

Chapter 7

Final Conclusions and Future Work

This thesis examined the problem of Colour Constancy in scenes with Varying Illumination. Many traditional algorithms have approached the Colour Constancy problem by trying to recover a single illuminant colour from an image. This is because their image formation model assumes that illumination colour is spatially constant across a scene. The field has a comparatively small amount of literature which tries to simultaneously solve for multiple lights in a scene. Finlayson et al. (1995) had shown that illumination is highly constrained in chromaticity space, and this is actually very advantageous in trying to constrain solutions to the Colour Constancy problem.

One of the most successful Colour Constancy algorithms is Gamut Mapping (Forsyth, 1990). This uses the varying surfaces in a scene to constrain the set of plausible mappings which could map an image rendering to some canonical illuminant. We have shown that our work can be unified with Gamut Mapping well, such that multiple illuminants can be used to further constrain the solution. The illuminant constraint is nearly always more powerful because illuminants in the world are naturally more predictable. Therefore they have a much bigger impact on reducing the size of the solution set. It is practically impossible to perceive very saturated red colours in very saturated blue lighting conditions. This is the crux of the Gamut

Mapping assumption. Gamut Mapping is often applied to entire images without consideration for the varying illuminants in that scene. We have shown performance is greatly improved by ensuring that input pixels are all produced by the same light. The pixels produced by a different light can then be included with our extension to the model.

To make these contributions practically useful, we needed to automatically detect areas of varying illumination in a scene. We have proposed a simple way of finding shadow edges in images, by exploiting the 1-dimensionality of illumination change. The pixels on either side of a shadow edge (associated with the same surface) can then be extracted for use with the algorithms we have proposed. We additionally have proposed a voting algorithm based on the previous work, which is much more robust to input errors, and can make use of the additional surface information. This means that we have a complete algorithm for the estimation of two white points in scenes with shadows. We have shown that the approaches described in this thesis are better than the current state of the art for illumination estimation in scenes with shadows on a variety of datasets.

For this work to have true practical application we need to calculate how many white points in a scene there are to estimate. Because we have used only colour information (rather than attempting to infer illumination change from scene geometry), then our detection method is likely to find shadow edge candidates in scenes with no shadows. This is because some surface changes could plausibly be modelled as illumination changes, and we would need higher level information to be able to truly distinguish these two cases. This would make a good area for further research, not only for it's applicability to Colour Constancy but also to scene understanding.

There are many other cues in a scene that could be useful towards illuminant estimation, such as specular highlights, inter-reflectance, understanding of geometry

and face detection. We know from human colour constancy, that the brain is using higher level information than bags of pixels to solve the problem. Future approaches in this field could produce algorithms that are tailored to the specific cues in a scene. Scene understanding is circularly related to colour constancy, as improvements in both areas will be respectively beneficial to each other. The work in this thesis has made contributions in this area by examining scenes with shadows (scene understanding) and being able to estimate multiple white points (colour constancy).

We believe the work in this thesis directs Colour Constancy research in a new direction, which can make more accurate predictions about the illuminants in images.

Bibliography

Abdel-Hakim, A. E. and Farag, A. A., 2006, "Csift: A sift descriptor with color invariant characteristics," in *Computer Vision and Pattern Recognition (CVPR)*, vol. 2, pp. 1978–1983, IEEE.

Alsabti, K., Ranka, S., and Singh, V., 1997, "An efficient k-means clustering algorithm," *Electrical Engineering and Computer Science*, , no. 43.

Barnard, K. and Finlayson, G., 1996, "Colour constancy for scenes with varying illumination," pp. 1–15, Springer Berlin Heidelberg.

Barnard, K., Martin, L., Funt, B., and Coath, A., 2002, "A data set for color research," *Color Research & Application*, vol. 27, no. 3, pp. 147–151.

Barrow, H. G. and Tenenbaum, J. M., 1978, *Computer Vision Systems*, chap. Recovering intrinsic scene characteristics from images, Academic Press.

Bianco, S. and Schettini, R., 2012, "Color constancy using faces," in *Computer Vision and Pattern Recognition (CVPR)*, pp. 65–72, IEEE.

Brainard, D. H. and Freeman, W. T., 1997, "Bayesian color constancy," *Journal of the Optical Society of America (JOSA)*, vol. 14, no. 7, pp. 1393–1411.

Buchsbaum, G., 1980, "A spatial processor model for object colour perception," *journal of the Franklin institute*, vol. 310, no. 1, pp. 1–26.

Canny, J., 1986, "A computational approach to edge detection," *Pattern Analysis and Machine Intelligence (PAMI)*, , no. 6, pp. 679–698.

Cardei, V. C., Funt, B., and Barnard, K., 2002, "Estimating the scene illumination chromaticity by using a neural network," *Journal of the Optical Society of America (JOSA)*, vol. 19, no. 12, pp. 2374–2386.

Ciurea, F. and Funt, B., 2003, "A large image database for color constancy research," in *Eleventh Colour Imaging Conference (CIC)*, pp. 160–163.

Comaniciu, D. and Meer, P., 2002, "Mean shift: A robust approach toward feature space analysis," *Pattern Analysis and Machine Intelligence (PAMI)*, vol. 24, no. 5, pp. 603–619.

Drew, M. S., Finlayson, G. D., and Hordley, S. D., 2003, "Recovery of chromaticity image free from shadows via illumination invariance," in *IEEE Workshop on Color and Photometric Methods in Computer Vision, ICCV*, pp. 32–39.

DZmura, M., 1992, "Color constancy: surface color from changing illumination," *Journal of the Optical Society of America (JOSA)*, vol. 9, no. 3, pp. 490–493, ISSN 1084-7529.

DZmura, M. and Iverson, G., 1993a, "Color constancy. i. basic theory of two-stage linear recovery of spectral descriptions for lights and surfaces," *Journal of the Optical Society of America (JOSA)*, vol. 10, no. 10, pp. 2148–2165, ISSN 1084-7529.

DZmura, M. and Iverson, G., 1993b, "Color constancy. ii. results for two-stage linear recovery of spectral descriptions for lights and surfaces," *Journal of the Optical Society of America (JOSA)*, vol. 10, no. 10, pp. 2166–2180, ISSN 1084-7529.

DZmura, M. and Iverson, G., 1994, "Color constancy. iii. general linear recovery of spectral descriptions for lights and surfaces," *Journal of the Optical Society of America (JOSA)*, vol. 11, no. 9, pp. 2389–2400.

Finlayson, G., 1996, "Color in perspective," *Pattern Analysis and Machine Intelligence (PAMI)*, vol. 18, no. 10, pp. 1034–1038.

Finlayson, G., Drew, M., and Funt, B., 1994, "Spectral sharpening: sensor transformations for improved color constancy," *Journal of the Optical Society of America (JOSA)*, vol. 11, no. 5, pp. 1553–1563.

Finlayson, G. and Hordley, S., 1998, "A theory of selection for gamut mapping colour constancy," in *Computer Vision and Pattern Recognition (CVPR)*, pp. 60–65, IEEE.

Finlayson, G., Hordley, S., and Morovic, P., 2005a, "Chromagenic colour constancy," in *International Colour Association (AIC)*, pp. 547–550.

Finlayson, G. D., 2013, "Corrected-moment illuminant estimation," in *International Conference on Computer Vision (ICCV)*, pp. 1904–1911.

Finlayson, G. D., Drew, M. S., and Lu, C., 2009, "Entropy minimization for shadow removal," *International Journal of Computer Vision (IJCV)*, vol. 85, no. 1, pp. 35–57.

Finlayson, G. D., Fredembach, C., and Drew, M. S., 2007, "Detecting illumination in images," in *International Conference on Computer Vision (ICCV)*, pp. 1–8, IEEE.

Finlayson, G. D., Funt, B. V., and Barnard, K., 1995, "Color constancy under varying illumination," in *International Conference on Computer Vision (ICCV)*, pp. 720–725, IEEE.

Finlayson, G. D., Hordley, S. D., and Drew, M. S., 2002a, "Removing shadows from images," in *European Conference on Computer Vision (ECCV)*, vol. 2353, pp. 823–836, Springer.

Finlayson, G. D., Hordley, S. D., and Hubel, P. M., 2002b, "Color by correlation: A simple, unifying framework for color constancy," *Pattern Analysis and Machine Intelligence (PAMI)*, vol. 23, no. 11, pp. 1209–1221.

Finlayson, G. D., Hordley, S. D., and Morovic, P., 2005b, "Chromagenic filter design," in *International Colour Association (AIC)*, pp. 1079–1083.

Finlayson, G. D. and Trezzi, E., 2004, "Shades of gray and colour contancy," in *Twelfth Colour Imaging Coference (CIC)*, pp. 37–41.

Forsyth, D. A., 1990, "A novel algorithm for color constancy," *International Journal of Computer Vision (IJCV)*, vol. 5, no. 1, pp. 5–35.

Foster, D. H., Amano, K., Nascimento, S., and Foster, M. J., 2006, "Frequency of metamerism in natural scenes," *Journal of the Optical Society of America (JOSA)*, vol. 23, no. 10, pp. 2359–2372.

Fredembach, C. and Finlayson, G., 2008, "Bright chromagenic algorithm for illuminant estimation," *Journal of Imaging Science and Technology*, vol. 52.

Funt, B. V., Barnard, K., and Martin, L., 1998, "Is machine colour constancy good enough?" in *European Conference on Computer Vision (ECCV)*, pp. 445–459, Springer.

Funt, B. V. and Drew, M. S., 1988, "Color constancy computation in near-mondrian scenes using a finite dimensional linear model," in *Computer Vision and Pattern Recognition (CVPR)*, pp. 544–549, IEEE.

Funt, B. V., Drew, M. S., and Ho, J., 1991, "Color constancy from mutual reflection," *International Journal of Computer Vision (IJCV)*, vol. 6, no. 1, pp. 5–24.

Funt, B. V. and Finlayson, G. D., 1995, "Color constant color indexing," *Pattern Analysis and Machine Intelligence (PAMI)*, vol. 17, no. 5, pp. 522–529.

Gehler, P. V., Rother, C., Blake, A., Minka, T., and Sharp, T., 2008, "Bayesian color constancy revisited," in *Computer Vision and Pattern Recognition (CVPR)*, pp. 1–8.

Gershon, R., Jepson, A., and Tsotsos, J., 1988, "From [r, g, b] to surface reflectance: computing color constant descriptors in images," *Perception*, vol. 17, pp. 755–758.

Gevers, T. and Smeulders, A. W. M., 1999, "Color-based object recognition," *Pattern Analysis and Machine Intelligence (PAMI)*, vol. 32, no. 3, pp. 453–464.

Gijsenij, A. and Gevers, T., 2007, "Color constancy using natural image statistics," in *IEEE Transactions on Image Processing*, pp. 2475–2489.

Gijsenij, A., Gevers, T., and van de Weijer, J., 2009, "Physics-based edge evaluation for improved color constancy," in *Computer Vision and Pattern Recognition (CVPR)*, pp. 581–588.

Gijsenij, A., Gevers, T., and van de Weijer, J., 2010, "Generalized gamut mapping using image derivative structures for color constancy," *International Journal of Computer Vision (IJCV)*, vol. 86, no. 2, pp. 127–139.

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

Gijsenij, A., Lu, R., and Gevers, T., 2012, "Color constancy for multiple light sources," *IEEE Transactions on Image Processing*, vol. 21, no. 2, pp. 697–707.

Henderson, S. T. and Hodgkiss, D., 2002, "The spectral energy distribution of daylight," *British Journal of Applied Physics*, vol. 14, no. 3, p. 125.

Hernández-Andrés, J., Romero, J., Lee Jr, R. et al., 2001, "Colorimetric and spectroradiometric characteristics of narrow-field-of-view clear skylight in granada, spain," *Journal of the Optical Society of America (JOSA)*, vol. 18, no. 2, pp. 412–420.

Hordley, S., 2006, "Scene illuminant estimation: past, present, and future," *Color Research and Application*, vol. 31, no. 4, pp. 303–314.

Hordley, S. D. and Finlayson, G. D., 2004, "Re-evaluating colour constancy algorithms," in *International Conference on Pattern Recognition (ICPR)*, vol. 1, pp. 76–79, IEEE.

Judd, D. B., MacAdam, D. L., Wyszecki, G., Budde, H. W., Condit, H. R., Henderson, S. T., and Simonds, J. L., 1964, "Spectral distribution of typical daylight as a

function of correlated color temperature," *Journal of the Optical Society of America (JOSA)*, vol. 54, no. 8, pp. 1031–1040.

Kawakami, R. and Ikeuchi, K., 2009, "Color estimation from a single surface color," in *Computer Vision and Pattern Recognition (CVPR)*, pp. 635–642, IEEE.

Kawakami, R., Ikeuchi, K., and Tan, R., 2005, "Consistent surface color for texturing large objects in outdoor scenes," in *Digitally Archiving Cultural Objects*, vol. 2, pp. 279–295.

Lagarias, J. C., Reeds, J. A., Wright, M. H., and Wright, P. E., 1998, "Convergence properties of the nelder-mead simplex method in low dimensions," *SIAM Journal on Optimization*, vol. 9, no. 1, pp. 112–147.

Land, E., 1974, "The retinex theory of color vision," *Scientific America*, , no. 47, pp. 23–58.

Land, E. and McCann, J., 1971, "Lightness and retinex theory," *Journal of the Optical Society of America (JOSA)*, vol. 61, no. 1, pp. 1–11.

Lee, H.-C., 1986, "Method for computing the scene-illuminant chromaticity from specular highlights," *Journal of the Optical Society of America (JOSA)*, vol. 3, no. 10, pp. 1694–1699.

Lowe, D. G., 1999, "Object recognition from local scale-invariant features," in *International Conference on Computer Vision (ICCV)*, vol. 2, pp. 1150–1157, Ieee.

M Bleier, M., Riess, C., Beigpour, S., Eibenberger, E., Angelopoulou, E., Troger, T., and Kaup, A., 2011, "Color constancy and non-uniform illumination: Can existing algorithms work?" in *Computer Vision Workships at the International Conference on Computer Vision (ICCV)*, pp. 774–781, IEEE.

Maloney, L. T. and Wandell, B. A., 1986, "Color constancy: a method for recovering surface spectral reflectance," *Journal of the Optical Society of America (JOSA)*, vol. 3, no. 1, pp. 29–33.

Mikolajczyk, K. and Schmid, C., 2004, "Scale and affine invariant interest point detectors," *International Journal of Computer Vision (IJCV)*, vol. 60, no. 1, pp. 63–86.

Ning, W., De, X., and Bing, L., 2009, "Edge-based color constancy via support vector regression," *IEICE transactions on information and systems*, vol. 92, no. 11, pp. 2279–2282.

Panagopoulos, A., Vicente, T. F., and Samaras, D., 2011, "Illumination estimation from shadow borders," in *International Conference on Computer Vision (ICCV)*, pp. 798–805, IEEE.

Parkkinen, J. P., Hallikainen, J., and Jaaskelainen, T., 1989, "Characteristic spectra of munsell colors," *Journal of the Optical Society of America (JOSA)*, vol. 6, no. 2, pp. 318–322.

Rosenberg, C., Minka, T., and Ladsariya, A., 2003, "Bayesian color constancy with non-gaussian models," *Advances in Neural Information Processing Systems*, vol. 16.

Rousseeuw, P. J., 1984, "Least median of squares regression," *Journal of the American statistical association*, vol. 79, no. 388, pp. 871–880.

Sapiro, G., 1999, "Color and illuminant voting," *Pattern Analysis and Machine Intelligence (PAMI)*, vol. 21, no. 11, pp. 1210–1215.

Shafer, S. A., 1985, "Using color to separate reflection components," *Color Research & Application*, vol. 10, no. 4, pp. 210–218.

Shi, L. and Funt, B., 2009, "Re-processed version of the gehler color constancy dataset of 568 images," http://www.cs.sfu.ca/ colour/data/.

Skaff, S., Arbel, T., and Clark, J. J., 2009, "A sequential bayesian approach to color constancy using non-uniform filters," *Computer Vision and Image Understanding*, vol. 113, no. 9, pp. 993–1004.

Stark, H., Yang, Y., and Yang, Y., 1998, *Vector space projections: A numerical approach to signal and image processing, neural nets, and optics*, John Wiley & Sons, Inc. New York, NY, USA, ISBN 0471241407.

Swain, M. J. and Ballard, D. H., 1991, "Color indexing," *International Journal of Computer Vision (IJCV)*, vol. 7, no. 1, pp. 11–32.

Tan, R. T., Nishino, K., and Ikeuchi, K., 2004, "Color constancy through inverse-intensity chromaticity space," *Journal of the Optical Society of America (JOSA)*, vol. 21, no. 3, pp. 321–334.

Tominaga, S., 1996, "Surface reflectance estimation by the dichromatic model," *Color Research & Application*, vol. 21, no. 2, pp. 104–114.

Tominaga, S. and Wandell, B. A., 1989, "Standard surface-reflectance model and illuminant estimation," *Journal of the Optical Society of America (JOSA)*, vol. 6, no. 4, pp. 576–584.

Tsin, Y., Collins, R. T., Ramesh, V., and Kanade, T., 2001, "Bayesian color constancy for outdoor object recognition," in *Computer Vision and Pattern Recognition (CVPR)*, vol. 1, pp. I–1132, IEEE.

Veksler, O., Boykov, Y., and Mehrani, P., 2010, "Superpixels and supervoxels in an energy optimization framework," in *European Conference on Computer Vision (ECCV)*, pp. 211–224, Springer.

Viola, P. and Jones, M., 2001, "Rapid object detection using a boosted cascade of simple features," *Computer Vision and Pattern Recognition (CVPR)*, vol. 1, pp. I–511.

Weijer, J. and Gevers, T., 2007, "Edge-based colour constancy," *IEEE Trans. on Image Processing*, URL `http://www.narcis.info/publication/RecordID/oai:uva.nl:285872`.

Weiss, Y., 2001, "Deriving intrinsic images from image sequences," in *International Conference on Computer Vision (ICCV)*, vol. 2, pp. 68–75, IEEE.

Wyszecki, G. and Stiles, W. S., 1982, *Color science*, Wiley New York.

Xiong, W. and Funt, B., 2006, "Estimating illumination chromaticity via support vector regression," *Journal of Imaging Science and Technology*, vol. 50, no. 4, pp. 341–348.