Color Constancy for RGB and Multispectral Images

by

Xiaochuan Chen

B.Eng., Beijing University of Posts and Telecommunications, 2013

Thesis Submitted in Partial Fulfillment of the
Requirements for the Degree of
Master of Science

in the
Department of Computing Science
Faculty of Applied Sciences

SIMON FRASER UNIVERSITY
Spring 2017

Approval

Name:	**Xiaochuan Chen**
Degree:	**Master of Science (Computer Science)**
Title:	***Color Constancy for RGB and Multispectral Images***
Examining Committee:	**Chair:** Jiannan Wang Assistant Professor

Mark S. Drew
Co-Senior Supervisor
Professor _____

Ze-Nian Li
Co-Senior Supervisor
Professor _____

Brian Funt
Examiner
Professor _____

Date Defended: _____

Abstract

The problem of inferring the light color for a scene is called Illuminant Estimation. This step forms the first task in many workflows in the larger task of discounting the effect of the color of the illuminant, which is called Color Constancy. Illuminant Estimation is typically used as a pre-processing step in many computer vision tasks. In this thesis, we tackle this problem for both RGB and multispectral images. First, for RGB images we extend a moments based method in several ways: firstly by replacing the standard expectation value, the mean, considering moments that are based on a Minkowski p-norm; and then secondly by going over to a float value for the parameter p and carrying out a nonlinear optimization on this parameter; and finally by considering a different expectation value, generated by using the geometric mean. We show that these strategies can drive down the median and maximum error of illuminant estimates. And then for multispectral images, we formulate a multiple-illuminants estimation problem as a Conditional Random Field (CRF) optimization task over local estimations. We then improve local illuminant estimation by incorporating spatial information in each local patch.

Keywords: Color Constancy, Corrected Moments, Illuminant Estimation, Multispectral Images, Multiple Illuminants

Acknowledgements

I want to use this opportunity to thank my supervisor Dr. Mark Drew and Dr. Ze-Nian Li for their support in both my researches and internships, their encouragement and patience let me experience different researches and works which are very valuable in my life. I want to express my great thanks to Dr. Mark Drew, without his guidance and invaluable insights, this work could not have been done. I would also like to thank Dr. Ze-Nian for many things he taught me during my graduate career.

I am also grateful to my committee for their help in my thesis defence: my examiner Brian Funt and the chair Jiannan Wang. I'd like to extend my thanks to Dr. Imari Sato and Dr. Yinqiang Zheng for their help during my internship in NII, and all my colleagues in Vision and Media Lab for helpful comments and discussions.

Table of Contents

Approval ii

Abstract iii

Acknowledgements iv

Table of Contents v

List of Tables vii

List of Figures viii

1 Introduction 1

1.1 RGB Illuminant Estimation 1

1.2 Multispectral Image Illuminant Estimation 2

2 Related Work 4

2.1 Gray-World and Gray-Edge 4

2.2 Extensions 6

2.3 Learning-Based Methods 6

2.4 Illuminant and Reflectance Spectra Separation 8

2.5 Conditional Random Field 9

3 Illuminant Estimation for RGB images 11

3.1 Corrected Moments 11

3.1.1 Polynomial Regression 11

3.1.2 Corrected Polynomial Regression 12

3.2 Corrected Moments with p-Norm 14

3.3 Optimization of p-Norm 15

3.4 Corrected Moments with Geometric Mean 15

4 Illuminant Estimation for Multispectral Images 17

4.1 Rank Matrix Factorization 17

4.2 Improved Local Estimation 19

4.3 Local Illuminant Search 20
4.4 Different Illuminant Regions Segmentation via CRF 21

5 Experiments **23**
5.1 RGB Domain Experiments 23
5.2 Multispectral Domain Experiments 26

6 Conclusion and Future Work **30**

Bibliography **31**

List of Tables

Table 4.1 32-by-32 patches' illumination and reflectance separation performance comparison between our method with [53], here is the reflectance estimation error 20

Table 5.1 Performance on the reprocessed [38] Color Checker Dataset [42]. Here we use angular error [22] to represent the performance for each method. Asterisk denotes our reproduced result of method [20] 24

Table 5.2 Performance on the reprocessed [49] Color Checker Dataset [42]. Here we use angular error [22] to represent the performance for each method. Asterisk denotes our reproduced result of method [20] 25

Table 5.3 Performance on the GrayBall dataset, followed [28] pre-processing steps. Asterisk denotes our reproduced result of method [20] 26

Table 5.4 Performance on the SFU Object dataset[6]. Asterisk denotes our reproduced result of method [20] 27

List of Figures

Figure 1.1 Taken from wikipedia https://en.wikipedia.org/wiki/Color_balance, the left half shows the photo as it came from the digital camera. The right half shows the photo adjusted to make a gray surface neutral in the same light. 2

Figure 2.1 Taken from [54],this example shows the limits of gray world assumption. Once the image contains little color variance, then the algorithm result will be biased 4

Figure 2.2 Taken from [28], three acquisitions of the same scene under different light sources. On the bottom line the color derivative distributions are shown, where the axes are the opponent color derivatives and the surfaces indicate derivative values with equal occurrence and darker surfaces indicating a more dense distribution. Note the shift of the orientation of the distribution of the derivatives with the changing of the light source. 5

Figure 2.3 Taken from [40], this is the general process for most learning based algorithms. Firstly, the feature or corse estimation was extracted, then it learns the weights to fuse the feature and coarse estimation. During the test phase, the learned weights was applied to the process and outputs the final estimation. These weights can either be specified to be hard (one illuminants was selected) or soft (a weighted average of estimates). 7

Figure 2.4 Taken from [31], it first queries the K nearest neighbours for each surface in the image, then the illuminant was estimated based on all the neighbouring models. 8

Figure 2.5 Taken from [7], these images are the same except for "tints" the scaling of red and blue. The bottom line is their log-chrominance histograms (with an axis overlaid for easier visualization, horizontal = u, vertical = v). Tinting an image affects the image's histogram only by a translation in log-chrominance space. This observation enables their convolutional approach to color correction, in which the algorithm learns to localize a histogram in this 2D space. . . . 9

Figure 2.6 Taken from [40] **a**. In training stage, illuminants are clustered into K classes. **b**. The network was trained to predict the correct class of the illuminant. **c**. During in test phase, firstly the network predict each class's score, then weighted average for each class was used as the final estimation 10

Figure 5.1 Approximated CIELAB error. We first estimated the illuminants for the image, then compute the image reflectance by diving the estimated illuminants. For the reflectance, we first transform it into CIELAB color domain and then calculate the error with the ground-truth. The Moments and Edges columns are from [20]. The last 4 columns are our results. 28

Figure 5.2 All the displaying image are hyper-spectral images, we convert them to RGB image for better display. (a) is the original reflectance image, (b) is the reflectance image illuminated by 4000K daylight on the left half and 8000K daylight on the right half, (c) is the way we divide image into patches, each patch size is 32-by-32, (d) is showing the illumination's index ranging from 1 to 56, (e) is the binary classification result when use index 30 as threshold to segment the image, (f) is the segmented different illuminated regions using graph cut based on local estimation 29

Figure 5.3 (a) is the original reflectance image, (b) is the recovered reflectance image based on method [17] which assume one illumination in the scene, (c) is the recovered reflectance using our method, (d) is recovered illumination using method [17], (e) is recovered illumination using our method 29

Chapter 1

Introduction

1.1 RGB Illuminant Estimation

Colors in images result from the combination of illuminant, surface reflection, and camera sensors plus the effects of the imaging and display pipeline [43]. In general, the human visual system is capable of filtering out the effects of the illuminant source when observing a scene – a psychophysical phenomenon denoted color constancy (CC). In many computer vision or image processing problems, researchers have often made use of some variety of CC as a pre-processing step to either generate data that is relatively invariant to the illuminant, or on the other hand to ensure that the captured color of the scene changes appropriately for different illuminant conditions. The computer science goal in the color constancy task is to estimate the illuminant, or at least the chromaticity – color without magnitude. We illustrated the general problem for color constancy in Fig.1.1.

Remarkably, the recent Corrected Moments illuminant estimation due to Finlayson [20] was reported to do overall best in terms of illuminant accuracy, and moreover produces results that reduce the maximum error in estimation. The latter property is important and desired: a camera manufacturer wishes to generate no images at all that produce strange colors, in any situation. The objective we aim at, here, falls within the scenario of a camera company (or smartphone producer) providing a CC algorithm with their equipment. In this sense, a training phase would be acceptable since the resulting algorithm adheres only to a single camera – the images we consider are not "unsourced" in the sense that come from the web or other unknown source: instead, they come from a known camera.

In this thesis we re-examine Finlayson's Corrected Moments method [20] with a view to simple extensions which we find further improve the illuminant estimates delivered by the method. These simple extensions do not greatly affect the good time- and space-complexity of the method, yet yield better results, surpassing some of the best results to date.

Here we extend the Corrected-Moments approach in three ways. Specifically, we begin by incorporating Minkowski-norm moments into Corrected-Moments illuminant estimation.

Then we show how to incorporate the Zeta-Image [17] approach to illuminant estimation within the Corrected-Moments method. Finally we devise a float-parameter optimization scheme to deliver the best performance for each dataset situation.

1.2 Multispectral Image Illuminant Estimation

Similar to RGB image, spectral distributions in hyperspectral images result from the product of the illuminant spectra, surface reflection, and the effects of camera sensors' band sensitivities. Separating illuminant and reflectance signals from an observed hyper-spectral image has been a longstanding problem in computer vision [14, 16, 26].

In general, two main streams have been used in solving this problem. The first is the use of hyper-spectral image statistics information to estimate illuminant and then separate reflectance information from the image. Another stream is to separate illuminant and reflectance directly from the observed hyper-spectral signal. This process is under-constrained, since the goal consists of twice the number variables as observations. A typical

Figure 1.1: Taken from wikipedia https://en.wikipedia.org/wiki/Color_balance, the left half shows the photo as it came from the digital camera. The right half shows the photo adjusted to make a gray surface neutral in the same light.

way to resolve this issue is by introducing a low dimensional subspace for reflectance or illuminant or both. Remarkably, [53] addresses a more general scenario where surface reflectance is assumed to fall into a low dimensional subspace while no constraint is placed on illuminant. In this way, the recovered illuminant would contain much more information compared to single RGB estimation.

In [53], the separation problem is modelled as a low-rank matrix factorization, with disentangled illuminant and reflectance found by solving a small convex quadratic program. Their method is robust even when the scene is under general, non-smooth, illuminant. However, it failed if their exist dominant shadows or multiple illuminants in the scene.

In this thesis, we improved Zheng's method [53] with incorporating spatial information into optimization step. Moreover, we introduce a CRF (Conditional Random Field) model over the results of separations from method [53] to handle scenes containing multiple illuminants or dominant shadows.

Chapter 2

Related Work

2.1 Gray-World and Gray-Edge

In experiments and tables of results below, note that we obtain results comparable to the best to date, state-of-the-art methods. However, in fact the method in [20] is based on very simple algorithms, so we begin the discussion with these. The simplest illuminant estimation algorithm is the Gray-World algorithm [11], which assumes that the average reflectance in a scene is achromatic. Thus the illuminant color may be estimated by simply taking the global average over pixel values. More specifically, in each color channel $k = 1..3$, the gray-world estimate of light color is given by $E(R_k)$, where $E(\cdot)$ is expectation value and R_k is RGB color. That is, Gray-World states that $E(R_k) = \frac{1}{N}\sum_{k=1}^{N} R_k$, with N being the number of pixels. Intuitively, Gray-World will obviously fail if a scene is insufficiently colourful. For example, an image of a gold coin that takes up most of the pixels will generate a very poor illuminant estimate; and if we move the white point to $R = G = B = 1$, or use a more careful white-point camera balance (e.g., [36]) then our image will likely end up containing a coin that looks gray rather than gold. See Fig.2.1

(a) Correct Balanced (b) Gray World Recovered

Figure 2.1: Taken from [54],this example shows the limits of gray world assumption. Once the image contains little color variance, then the algorithm result will be biased

A more recent but almost as simple algorithm is the Gray-Edge method, which asserts that the average of reflectance differences in a scene is achromatic [28]. With this assumption, the illuminant color is estimated by computing the average color derivative in the image, $E(||\nabla R_k||)$, where ∇ is the gradient field pair $\{\partial/\partial x, \partial/\partial y\}$. The Gray-Edge assumption originated from the empirical observation that the color derivative probability distribution for images forms a relatively regular, ellipsoid-like shape, with the long axis coinciding with the illuminant color [28]. The expectation value for the kth color channel is then estimated by

$$\hat{c}_k = \sqrt{\sum_{i=1}^{N} \left| \frac{\partial R_k^i}{\partial x} \right|^2 + \left| \frac{\partial R_k^i}{\partial y} \right|^2}, \quad k = 1..3 \tag{2.1}$$

with $\hat{c}_k$ denoting the estimate of the illuminant color.

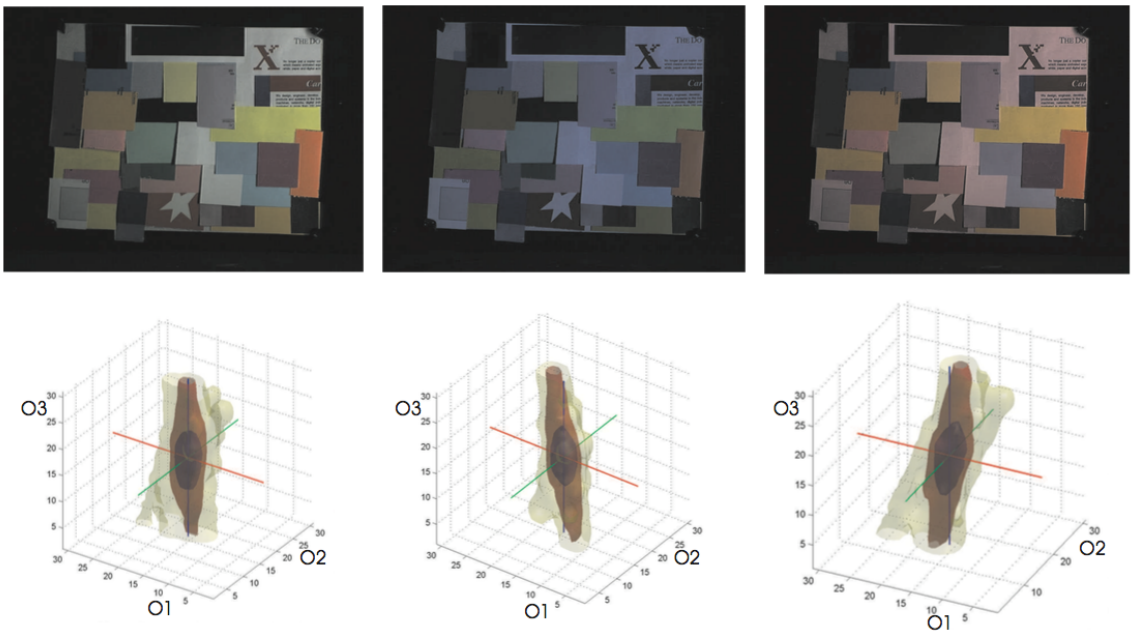


Figure 2.2: Taken from [28], three acquisitions of the same scene under different light sources. On the bottom line the color derivative distributions are shown, where the axes are the opponent color derivatives and the surfaces indicate derivative values with equal occurrence and darker surfaces indicating a more dense distribution. Note the shift of the orientation of the distribution of the derivatives with the changing of the light source.

Based on the Gray-World and Gray-Edge approaches, many extensions have been proposed. The Shades-of-Gray algorithm [21] was the first to propose using a Minkowski norm to replace the averaging step in the Gray-World method. With integer exponent p, the Minkowski or p-norm is $\sqrt[p]{\sum_{i=1}^{N} |R_k|^p}$. Similarly, applying the Minkowski norm to the Gray-Edge method substantially improves illuminant estimation performance [21]. This

estimate is as follows:

$$\hat{c}_k = \sqrt[p]{\sum_{i=1}^{N} \left| \frac{\partial R_k^i}{\partial x} \right|^p + \left| \frac{\partial R_k^i}{\partial y} \right|^p}, \quad k = 1..3 \tag{2.2}$$

Note that these methods are based on moments of the pixel values or of the gradient fields.

2.2 Extensions

Starting from the Gray-Edge method, it was also found that first blurring the image with a Gaussian smoothing filter could aid performance. And as another extension, considering higher-order derivatives of order n could sometimes help as well. So, replacing the partial derivatives with Gaussian derivatives denoted δ_σ, Gray-Edge becomes

$$\hat{c}_k = \sqrt[p]{\sum_{i=1}^{N} |\frac{\delta_\sigma^n R_k^i}{\delta_\sigma x^n}|^p + |\frac{\delta_\sigma^n R_k^i}{\delta_\sigma y^n}|^p} \tag{2.3}$$

where σ indicates the standard deviation in a Gaussian derivative and n is the order of derivative.

Ref.[5] presents a useful review of various CC algorithms, some of which are quite complex. In eq. (2.3) the order n determines whether the method is Gray-World, with $n = 0$ (i.e., using RGB pixel values, not gradient fields), or Gray-Edge, with $n \geq 1$; p denotes the Minkowski norm; and σ is the parameter of Gaussian filter parameter for smoothing the original image; smoothing tends to improve results since it removes noise. The original Gray-World has $n = 0$ and $p = 1$; when $n = 0$ and $p \to \infty$ the algorithm is equivalent to the so-called Max-RGB approach, with illuminant estimate given by the maximum value in each of R, G, and B; and when $n = 0$ and $0 < p < \infty$ it is Shades of Gray[21]. An advantage of all the color constancy approaches based on eq. (2.3) is that they are of low computational complexity both in space and in time [28]. Moreover, these approaches do not require any training stage. However, the latter point may arguably be a disadvantage, from the point of view of a manufacturer willing to calibrate a particular camera for best estimates.[1]

2.3 Learning-Based Methods

First attempt is using neural networks [12] to train a model for predicting the chromaticity values of the illuminant with the input of binarized chromaticity histogram of the image. Though this approach may deliver good performance when trained properly, while it suffer from limited dataset when it was first introduced. Other similar approaches applied linear

[1] Saturated pixels are omitted for all these methods.

regression[3, 2, 1] or support vector regression [52, 23] to the same type of feature input. And all these methods could be generalized as in Fig.2.3

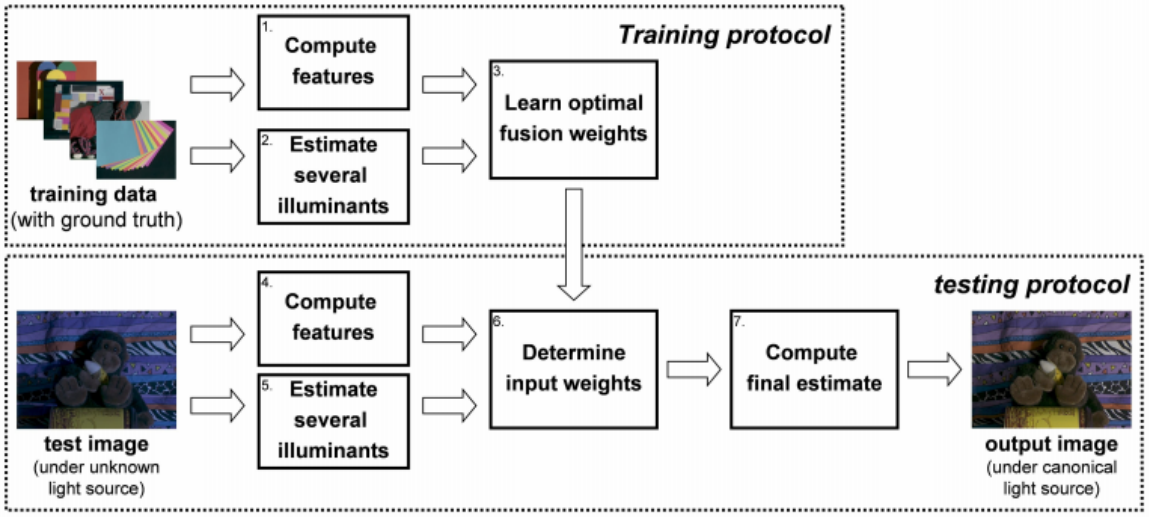


Figure 2.3: Taken from [40], this is the general process for most learning based algorithms. Firstly, the feature or corse estimation was extracted, then it learns the weights to fuse the feature and coarse estimation. During the test phase, the learned weights was applied to the process and outputs the final estimation. These weights can either be specified to be hard (one illuminants was selected) or soft (a weighted average of estimates).

Recently, more well designed learning based methods improved the performance a lot. The Examplar-Based [31] method treats an image as a composition of surfaces, and learned a representation model for training surfaces. During the testing phase, the method is finding K-nearest neighbour models and use these models to estimate the illuminant. Fig.2.4.

Ref [7] applied ideas from object detection and structured prediction to color constancy problem with the observation that scaling the color channel of the image will induce the translation in the log-chromaticity histogram of that image. And they formalized the color constancy problem into a discriminative learning problem. Fig.2.5

Deep learning shows amazing improvement in computer vision tasks, like image classification [34, 15, 50, 33], object detection [46, 30, 24, 48], and segmentation [27, 18] etc. Many attentions are paid to use deep learning to aid color constancy problem. Bianco et at. [9] trained a network based on AlexNet, they are using image patches as input and predict the local illuminant through the network. Doing so, they could have more data to train the network and reduce the computation for single estimation. Lou et al. address the data limits issue by first training the network using Imagenet dataset as training set, since this dataset doesn't contain the ground truth for the illuminant, [55] used gray world method to compute the illuminant. In this way, the trained network adapted well to illuminant estimation task and overcome the data limits problem. Finally they just fine-tuned the trained network to each test dataset domain. Oh et al [40] approach the color constancy

problem as a classification problem. And the classification scores were used to compute the final estimation of the illuminant. Detailed processes could be find in Fig.2.6

2.4 Illuminant and Reflectance Spectra Separation

Multispectral illuminant estimation, or illuminant and reflectance spectra separation (IRSS) can be summarized into two categories, either separating a single hyper-spectral image into illuminant and reflectance image, or with an eye to the colour constancy task of estimating illuminant for RGB images.

Classically, for this signal separation task, Ho et al. [26] first made use of the low dimension model for both illuminant and reflectance. To improve accuracy of the separation, Chang et al. [14] enforced more constraints on illuminant and reflectance. In the work [16], Drew and Finlayson successfully improved the bilinear computation's efficiency by moving to logarithmic space.

The color constancy problem mostly consists of estimating the illuminant, or at least the chromaticity of the light in the observed scene. In [45], Robles-Kelly et al. generalized the assumptions and methods in colour constancy specifically for the IRSS problem. In principle, many colour constancy methods, such as [11, 35, 44], could be carried out in a multispectral domain, provided their underlying assumptions were still satisfied.

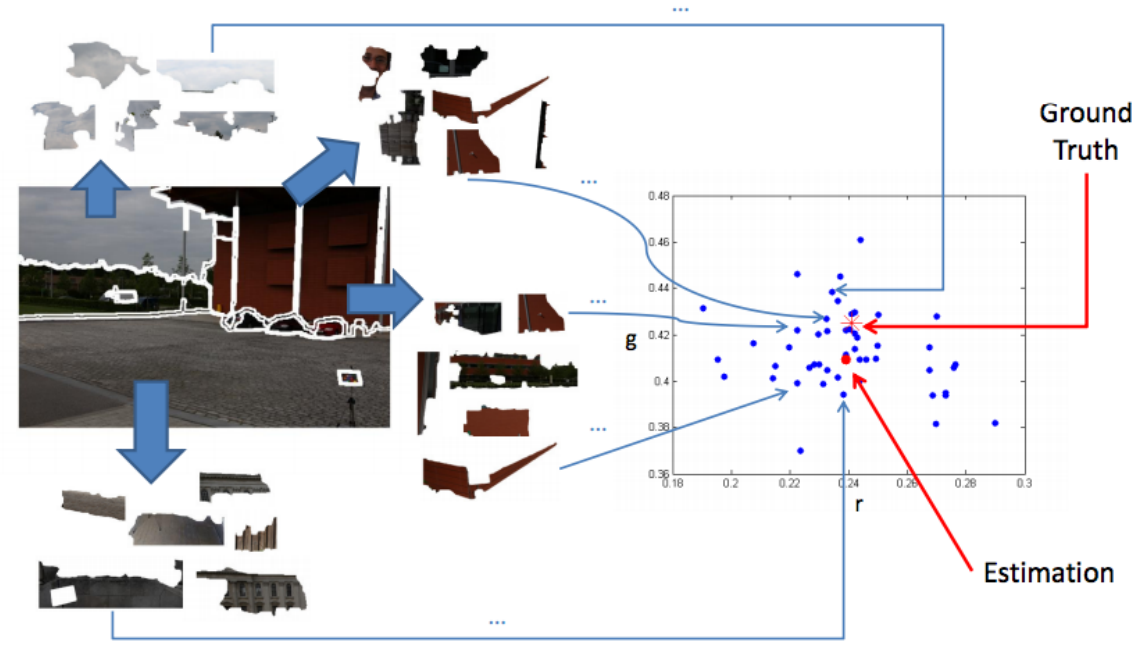


Figure 2.4: Taken from [31], it first queries the K nearest neighbours for each surface in the image, then the illuminant was estimated based on all the neighbouring models.

2.5 Conditional Random Field

Using a CRF model to incorporate the spatial information in a scene has been widely used in several works. Lu and Drew [37] applied a Markov Random Field (MRF) for shadow and non-shadow region separation. Their method first generated an illuminant invariant image by using a model guided by the assumption of Planckian illuminant. They formulated a pairwise potential which allowed them to label shadows smoothly. In [8], Beigpour and Riess et al. fused multiple physical and statistical based colour constancy methods into a CRF model which produced pixel-wise labels for shadows. However, their method relied on many complex colour constancy methods to generate accurate estimations for the CRF model, and this is time consuming process.

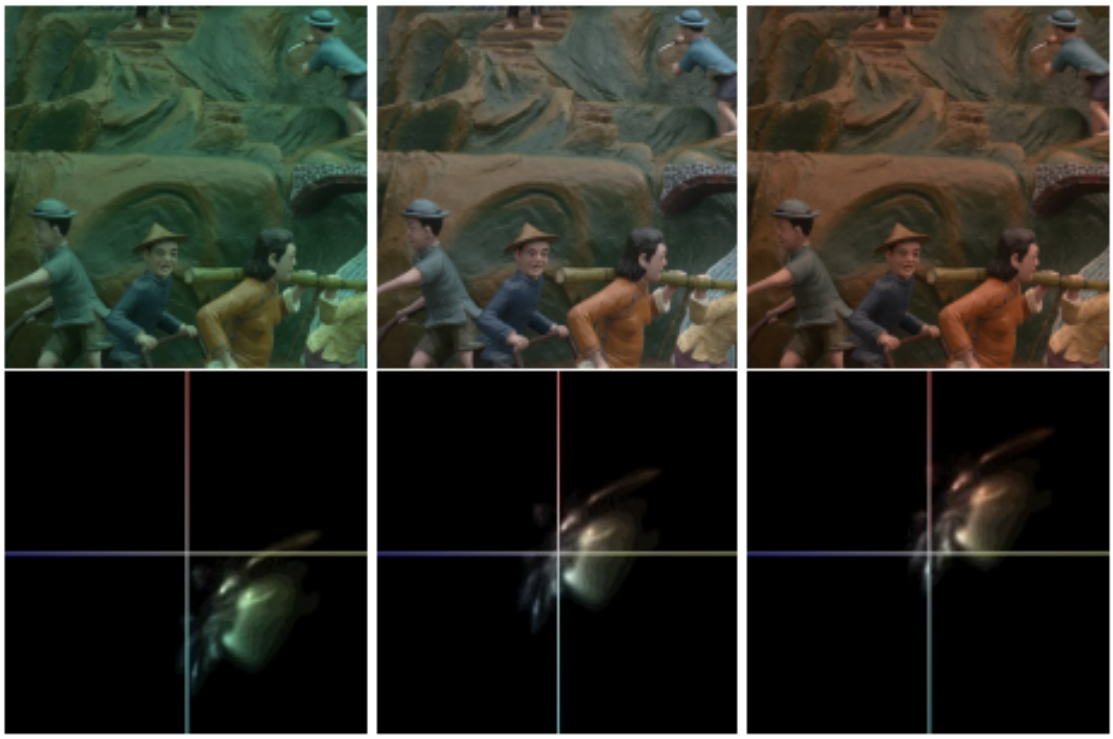


Figure 2.5: Taken from [7], these images are the same except for "tints" the scaling of red and blue. The bottom line is their log-chrominance histograms (with an axis overlaid for easier visualization, horizontal = u, vertical = v). Tinting an image affects the image's histogram only by a translation in log-chrominance space. This observation enables their convolutional approach to color correction, in which the algorithm learns to localize a histogram in this 2D space.

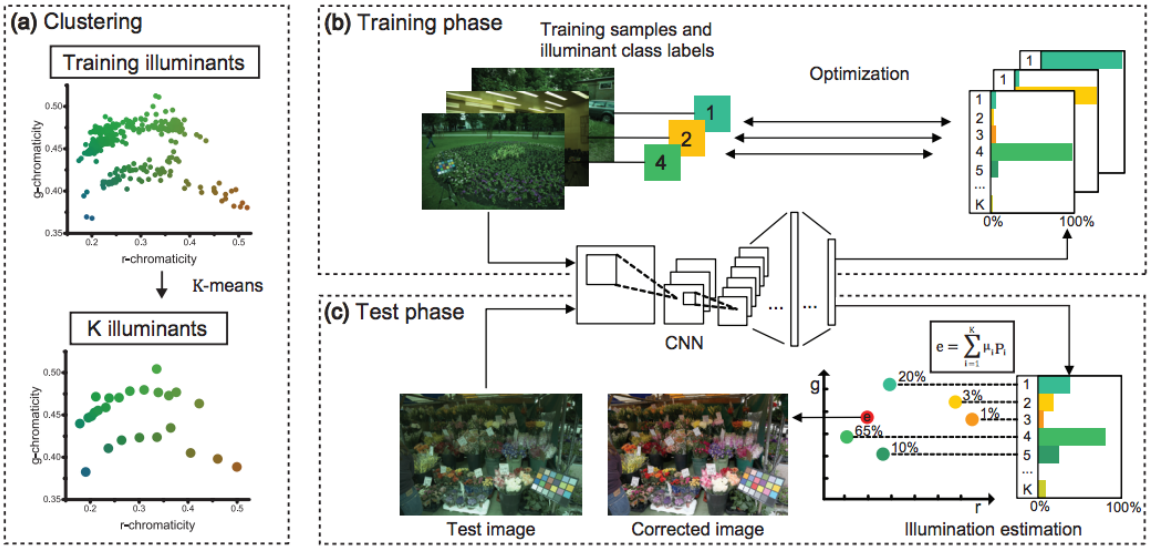


Figure 2.6: Taken from [40] **a**. In training stage, illuminants are clustered into K classes. **b**. The network was trained to predict the correct class of the illuminant. **c**. During in test phase, firstly the network predict each class's score, then weighted average for each class was used as the final estimation

Chapter 3

Illuminant Estimation for RGB images

3.1 Corrected Moments

To date, most recently the most successful descendant of the methods above is that in [20]. The method in [20] derives from Eq. (2.1), using either moments of RGB or moments of first derivatives of RGB. In that work it was found, remarkably, that by incorporating simple but fundamental corrections to the above moments-based approaches, i.e. the Gray-World and the original Gray-Edge methods, illuminant estimation performance bested state-of-art approaches involving a great deal of computation and even feature extraction (see [13, 20, 21, 29, 5, 31]).

In this thesis we mean to extend the Corrected-Moments approach, in several ways, and thus improve the performance even more. Our first extension is to replace a standard expectation value mean by instead using a Minkowski p-norm. Moreover we go over to a float value for the parameter p by carrying out a nonlinear optimization on this parameter. Secondly, we also consider a different expectation value generated by using the geometric mean, as in [17]. And thirdly, we also replace the corrected moments method use of gradient data as in Eq. (2.1) by instead using a Minkowski p-norm. We show that these strategies can drive down the median and maximum error of illuminant estimates.

3.1.1 Polynomial Regression

The Corrected-Moments method can be understood as a variant of polynomial regression. Here we briefly recapitulate the method [20], pointing out its simple but innovative use of fundamental observations to arrive at an excellently performing algorithm. Firstly, in obtaining estimates $\hat{c}_k$ of the illuminant color in our Eq. (2.1) and in Eq. (2.2), we cannot expect to recover the absolute intensity of the light. This is due to the fact that the light and surface interact multiplicatively in forming the color signal spectrum which enters the

camera. E.g., a pink light on white walls looks like a white light on pink walls – modulo calculations of the inter-reflections at the corners, which arguably may help the human visual system disambiguate the situation [10]. So in our calculations of $\hat{c}_k$, we should always convert to estimates of the chromaticity of the light source, not its absolute strength; i.e., we form chromaticity – color without magnitude. Hence in [20], the author explicitly separates the estimates of chromaticity and of intensity (the latter consisting of light-strength times the particular albedo at the current pixel).

Let us first consider ordinary least squares polynomial regression. Recall that our scenario is that of a camera manufacturer developing a method of recovering the lighting chromaticity in any image; therefore we can assume we have available a training set of RGB images, plus ground truth for the illuminant chromaticity triple as well, perhaps by imaging a color target or white patch in each image (and then masking off the white patch for training and estimation). Our training-set regression should supply us with regression coefficients a_j which will allow us to recover an estimate of illuminant chromaticity from any new image in a testing set (for this same camera). Here we shall make use of the standard, L_1-norm based 3-vector chromaticity $c_k = R_k / \sum_{j=1}^{3} R_j$, $k = 1..3$. Our training set consists of n images plus n ground-truth chromaticity triples $\mathbf{c}$. For any estimate of light color that we derive, $\hat{c}_k$, we agree to always go to chromaticity $\hat{c}_k / \sum_{j=1}^{3} \hat{c}_j$ to judge our accuracy.

Let the degree of the polynomial be d, with the independent data being expectation values over polynomials for the entire image. For example, for $d = 2$ (i.e., up to 2nd degree monomials in R,G,B), we take independent values to be $E(R^2), E(G^2), E(B^2), E(RG), E(RB), E(GB)$ (all monomials of degree 2) as well as $E(R), E(G), E(B)$ (all of degree 1) and also an offset term 1 of degree 0. Let us call these 10 image descriptors a 1×10 row-vector $\mathbf{m}$. For dependent variables we take the ground truth 1×3 row-vector $\mathbf{c}$. Suppose there are n images plus n ground truth chromaticity vectors. Then standard Least Squares (LS) based polynomial regression groups all training-set image descriptors into an $n \times 10$ matrix $\mathbf{M}$ and correct chromaticities into an $n \times 3$ matrix $\mathbf{C}$, with the sought regression coefficients a_j then delivered by the minimization

$$\min_A ||\mathbf{MA} - \mathbf{C}||_2 \tag{3.1}$$

where matrix $\mathbf{A}$ is the 10×3 collection of regression coefficients a_j.

3.1.2 Corrected Polynomial Regression

Now, in the first place, in keeping with our goal of not trying to map illuminant intensities we should not consider, and thus remove, the constant (offset) term from Eq.(3.1), leaving a dimension-9 problem. Moreover, that term changes the black point of the camera and we do not wish to do that. Now, in a main innovative insight [25] it was also recognized that if we do not wish to model intensity, then we should make our regression coefficients invariant to

the actual (multiplicative) level of intensity of the light. To do so, roots of degree d should be applied to the moments listed above, thus keeping the units of light unchanged no matter what the light level is. That is, instead of $E(RG)$, we should use $\sqrt{E(RG)}$, etc. In [25], this insight was utilized in the color correction problem – mapping RGB to tristimulus values XYZ. In [20] this same insight was further applied to the illuminant estimation problem. Note that the number of expectation values used is still 9 in the $d = 2$ case, because $\sqrt{E(R^2)} \neq E(\sqrt{R^2})$. Thus our set of corrected moments is $\{E(R), E(G), E(B), \sqrt{E(R^2)}, \sqrt{E(G^2)}, \sqrt{E(B^2)}, \sqrt{E(RG)}, \sqrt{E(RB)}, \sqrt{E(GB)}\}$. Similarly, root-monomials of degree 3 would be $\sqrt[3]{E(RGB)}$, etc.

The second main innovative insight that completes the corrected-moment method [20] is explicitly taking into account the fact that we have only chromaticity values to regress upon, not intensities. Suppose an overall albedo times light-intensity scalar is k_j, for each of the $j = 1..n$ images. Then remembering that the right-hand-side $n \times 3$ matrix $\mathbf{C}$ consists of chromaticities, i.e., color without magnitude, in [20] the author sets up an optimization

$$\min_{K,A} ||\mathbf{KMA} - \mathbf{C}||_2 \tag{3.2}$$

where $\mathbf{K}$ is a diagonal array of $j = 1..n$ constants k_j; here, for $d = 2$ for example, $\mathbf{K}$ is $n \times n$, $\mathbf{M}$ is $n \times 9$, $\mathbf{A}$ is 9×3, and $\mathbf{C}$ is $n \times 3$. A solution is given iteratively by solving for coefficient matrix $\mathbf{A}$ and then for constant vector $\mathbf{k} = diag(\mathbf{K})$ [20] (here using the $d = 2$, i.e. 9-component variant for illustration):

Algorithm: Corrected Moments
Initialize $\mathbf{K} = \mathbf{I}_{n \times n}$, *the unit matrix*;
for *MAXITN iterations* **do**
  $\mathbf{A} = (\mathbf{KM})^+\mathbf{C}$ where $^+$ is the Moore-Penrose pseudo-inverse: size is 9×3;
  for $j = 1..n$ *images* **do**
    $\mathbf{m}$ = row j of $\mathbf{M}$, size is 1×9;
    $\mathbf{c}$ = row j of $\mathbf{C}$, size is 1×3;
    $k_j = \mathbf{c}(\mathbf{mA})^+$, a scalar.;
  end
end

The work in [20] applies the above algorithm to RGB image data R, or alternatively to gradient data ∇R, and for degree up to $d = 3$ (i.e., 19 moments); that work carries out experiments using 3-fold cross validation, dividing any data set into thirds, randomly assigning image-index values in $j = 1..n$ and taking 2/3 of the data as training and 1/3 as testing images. Empirically, the advantages of the method in [20] are two-fold: a reduction in illuminant estimate error is observed, and also a reduction in the maximum error – the latter is important because manufacturers desire no perceptible outlier cases.

In [20], the author used RGB moments as well as higher order RGB moments. These moments can be summarized as follows:

$$m_{uvw} = \left[\frac{\sum_{i=1}^{N} R_i^u G_i^v B_i^w}{N} \right]^{1/d} \qquad u+v+w = d; \quad u,v,w \geq 0 \tag{3.3}$$

Here R,G,B is used as a shorthand for either RGB pixel value themselves or Gaussian first (in [20]) derivatives of the image, with monomials m_{uvw} denoting the moments, with a total of $3, 9, 19, 34$ moments when the polynomial degree is $d = 1, 2, 3, 4$. The moments in (3.3) scale with intensity. And this "intensity scaling" property is important for correcting illuminant estimation [20]: with this property, calculated chromaticities will be independent of intensity – and this is not the case for standard polynomial regression.

Now let us consider extensions of this method: here we make use of sets of chromaticity data, along with color RGB data as well as gradient data ∇R; we introduce use of the p-norm into corrected moments; and as well make use of a geometric mean instead of a sum-of-squares method that minimizes the mean. In an innovative step, we also show how the Minkowski p-norm values can be shifted into the float domain and optimized for any camera/dataset situation.

3.2 Corrected Moments with p-Norm

Here we extend the basic corrected moments method (5) by utilizing a Minkowski p-norm in place of the averaging norm $E(\cdot)$. Again, let the maximum polynomial degree be d: typically, we might use values $d \in 1..4$. Now incorporating the Minkowski norm in our illuminant estimate, we produce moments as follows: for the linear, d=1 moments we use

$$(m_k)_{(d=1)} = (\frac{1}{N} \sum_{i=1}^{n} (R_k^i)^p)^{1/p}, \quad k = 1..3 \tag{3.4}$$

i.e., a 3-vector of p-norm moments, $k = 1..3$ for R,G,B, formed by summing over all pixels $i = 1..n$. The *degree* d of each monomial is $d = 1$, i.e., R, G, or B entering the p-norm as linear terms that are then formed into a p-norm.

For the remaining moments, formed from monomials in R,G,B of up to degree d which is higher than 1, we use a sum over degree $j = 2..d$. So e.g. for $j = 2$ we need p-norm expectation values made from monomials of degree 2, i.e., from products R^2, G^2, B^2, RG, RB, GB, for a total dimension of moments $D = 6$. In keeping with the idea above of making the moments scale linearly with intensity, the expectation values would still require a square root applied.

For degree $j = 3$ we need p-norm expectation values for monomials RGB, R^2G, R^2B, R^3, G^2R, G^2B, G^3, B^2R, B^2G, B^3, for a total dimension of moments $D = 10$; and the expectation values would still require a cube root applied.

In general, for degree of monomial j the dimension D is $D =_{(3+j-1)} C_j$. That is, for cases $j = 2, 3, 4$ the number of monomials is $D = 6, 10, 15$. Altogether, if we want moments up to degree $D = 3$, say, then we collect three degree-1 monomials, six degree-2 monomials, and ten degree-3 monomials for a total dimension of moments equal to 19. For $d = 4$ we have $j = 1..4$ meaning a total of $3 + 6 + 10 + 15 = 34$ moments.

We can state this concept succinctly as follows:

Moments of up to Degree d**:**

for $j = 2..d$ *// if monomial degree is* $d = 2, 3, 4$ *then number of moments* D *is* ${}_{(3+d-1)}C_d$ *= 6,10,15 etc.*

do

  $indSets$ = unique combinations of RGB indices 1,2,3 to polynomial degree j ;

  for $q = 1 : D$ **do**

    $(m_k)_{(j,q)} = (\frac{1}{N} \sum_{i=1}^{n} prod(R_k^i(indSets(row\, q))^p)^{1/j})^{1/p}$;

    // $k = 1..3$;

    // $prod$ = product across columns of indSets

  end

end

3.3 Optimization of p-Norm

In the above, Minkowski parameter p is fixed. Recalling that we mean to carry out the best estimation for a particular camera only, we can optimize over p. We used Matlab's *fmincon* function, setting variable p as the only varying parameter and median error as our optimization goal. We randomized the image-index value for both training and testing sets, using three-fold cross validation. Unusually, however, for this optimization schema we kept the training set and testing sets' index vectors fixed. Though of course this might cause over-fitting for the particular index settings, we found that the optimal value of p was actually very close to re-randomizing image indexes in each iteration. Using a fixed randomization greatly reduced the time for finding an optimal parameter p (we checked on fully randomized training and test sets, of course).

3.4 Corrected Moments with Geometric Mean

In [17] it was argued that, based on fundamental physical principles regarding matte and specular contributions to pixel color, an excellent estimate of illuminant color could be determined by calculating the geometric mean (or 'Geo-Mean' for short), in each color channel. Here we therefore replace the ordinary, summation-based moments, in use so far, with such multiplicative-based moments.

Similar to eq. (3.3), our geometric mean moments may be summarized in the following equation:

$$m_{uvw} = \left[\left(\prod_{i=1}^{N} R_i^u G_i^v B_i^w \right)^{1/N} \right]^{1/d} \quad u+v+w=d \;\; u,v,w \geq 0 \tag{3.5}$$

Here m_{uvw} now denotes the new moments. When we set $d = 1$, m_{uvw} is simply the illuminant color triple. Note that in [17], the authors proposed using only the top 10% brightest pixels because only near-specular pixels obey their observed planar constraint rule used to determine the illuminant. Hence, in this paper we also borrow this idea of using only the top-10% brightest pixels to compute the moments.[1] When we set $d = 2, 3$, we end up with 9 and 19 moments.

By inspection, the computed moments scale with intensity; again, this “intensity scaling" property is important for correcting illuminant estimation [20].

[1] If the pixel was saturated, it was not used. Also it was assumed to be linear image.

Chapter 4

Illuminant Estimation for Multispectral Images

4.1 Rank Matrix Factorization

Here we briefly recapitulate the method [53], pointing out its innovative part that arrives at an excellent method. Firstly, for a pixel at spectral band i in a hyper-spectral image, its value d_i is proportional to the product of the illuminant l_i and surface reflectance r_i; that is,

$$d_i = l_i r_i,\ 1 \leq i \leq m, \tag{4.1}$$

Here, m denotes the number of spectral bands (which is 57 in all our experiments), and a vector accounting for camera gains has been omitted. As well, spectral sensitivity has been pre-corrected to be unity for all spectral bands $i = 1..m$.

For a hyper-spectral image with n pixels under spatially uniform illuminant, the intensity d_{ij} of the j-th pixel for the i-th spectral band is

$$d_{ij} = l_i r_{ij},\ 1 \leq i \leq m,\ 1 \leq j \leq n, \tag{4.2}$$

In matrix form this reads

$$\underbrace{\begin{bmatrix} d_{11} & \cdots & d_{1n} \\ \vdots & \ddots & \vdots \\ d_{m1} & \cdots & d_{mn} \end{bmatrix}}_{D_{m\times n}} = \underbrace{\begin{bmatrix} l_1 & & \\ & \ddots & \\ & & l_m \end{bmatrix}}_{L_{m\times m}} \underbrace{\begin{bmatrix} r_{11} & \cdots & r_{1n} \\ \vdots & \ddots & \vdots \\ r_{m1} & \cdots & r_{mn} \end{bmatrix}}_{R_{m\times n}} \tag{4.3}$$

The above system is under-constrained, since the observation matrix D has mn constraints but with $m(n+1)$ variables in the diagonal illuminant matrix L and the reflectance matrix

R taken together. In order to solve this equation, more constraints must be added to illuminant and reflectance.

Reflectance spectra can easily be shown to usually lie in a low dimensional linear subspace. Therefore, eq. (4.2) could be written in the form

$$D_{m\times n} = L_{m\times m}R_{m\times n} = L_{m\times m}B_{m\times s}C_{s\times n}, \tag{4.4}$$

where B and C denote the spectral bases and coefficients respectively, and s is the subspace dimensionality. In addition to the above physically meaningful factorization, it could also be factorized using partial singular value decomposition (SVD) as follows

$$D_{m\times n} = U_{m\times s}S_{s\times s}V^T_{n\times s} = UQQ^{-1}SV^T = (UQ)(Q^{-1}SV^T), \tag{4.5}$$

Here, Q is an arbitrary $s \times s$ invertible matrix. Comparing eqs. (4.4) with (4.5), the IRSS problem comes down finding a proper matrix Q such that

$$UQ = LB \tag{4.6}$$

Recall that B is the reflectance basis matrix learned via principal component analysis of a spectra dataset [39, 32]. So based on eq. (4.6), the illuminant spectrum L can be solved for by minimizing the following equation:

$$\min_{L,Q}||UQ - LB||^2_F, \text{ s.t. } l_i \geq \max_j d_{ij},\ l_z \leq \max_{i,j} d_{ij} \tag{4.7}$$

in which $l_i \geq \max_j d_{ij}, 1 \leq j \leq n, 1 \leq i \leq m$ form constraints that each illuminant factor should be greater than or equal to its corresponding observed signal, given eq. (4.2), since the reflectance r should be less than or equal to one. And $l_z \leq \max_{i,j} d_{ij}$ where z is the row index of the maximum value $\max_{i,j} d_{ij}$: it restricts the illuminant's absolute scale. Since eq. (4.7) is a simple convex quadratic program (QP), L and Q can be solved for by iteratively updating L and Q.

After obtaining L, the accuracy of separation can be further improved by minimizing the following cost functions on the original low rank model equation, eq. (4.4):

$$\begin{aligned} &\min_{L,C}||D - LBC||^2_F, \\ &\text{s.t. } 0 \leq (BC)_{ij} \leq 1,\ l_i \geq \max_j d_{ij},\ l_z \leq \max_{i,j} d_{ij} \end{aligned} \tag{4.8}$$

After optimizing this equation, the illuminant L is well resolved and reflectance spectra values R are equal to BC.

4.2 Improved Local Estimation

The previous work [53] can be understood as incorporating surface reflectance basis coherence into the separation problem; in other words, the low dimension property for surface reflectance helps to separate illuminant and reflectance. Here, we are trying to implement the method [53] for individual patches in an image. We found in fact that separation accuracy for different patches varies a good deal. When the patch contains many colours, the separation of illuminant and reflectance is reasonable good; but when colour diversity is small accuracy is diminished. This observation is consistent with the experiments in [53], in which separation accuracy increases with more colours.

After carefully reexamining the method [53], we understood that this problem is caused by eq. (4.5). In that equation, D is decomposed into U, S, V via SVD. So U is heavily dependent on D and could be noisy if D only contains small variation. For example we typically assume the reflectance subspace dimension s is 8 (chosen as the good tradeoff between expressive power and noise resistance), while on a small patch D could be well modelled only using only a 3-dimensional subspace; then U is noisy for the 5-D orthogonal subspace from SVD factorization.[1]

Once the decomposed U is noisy, the next eqs. (4.7), (4.8) would result in very poor separation. In order to resolve this issue of poor separation on local patches, we propose incorporating local spatial information into the final optimization eq. (4.8). Our novel suggestion is to change eq. (4.8) in the following way:

$$\begin{aligned} \min_{\mathbf{l},C} ||D - LBC||_F^2 + \lambda ||CG||_F^2, \\ \text{s.t. } 0 \leq (BC)_{ij} \leq 1,\ l_i \geq \max_j d_{ij},\ l_z \leq \max_{i,j} d_{ij} \end{aligned} \tag{4.9}$$

$$CG = \underbrace{\begin{bmatrix} c_{11} & \cdots & c_{1n} \\ \vdots & \ddots & \vdots \\ c_{m1} & \cdots & c_{mn} \end{bmatrix}}_{C_{m\times n}} \underbrace{\begin{bmatrix} & \vdots & \vdots & \vdots & \vdots & \\ \cdots & g_{n\_l} & g_{n\_u} & g_{n\_r} & g_{n\_b} & \cdots \\ & \vdots & \vdots & \vdots & \vdots & \end{bmatrix}}_{G_{n\times 4n}}$$

Here G is a $n \times 4n$ sparse matrix; g is a n-by-1 column vector such as $[\cdots, 1, -1, \cdots]^T$; λ is an empirical parameter which we chose as 0.05 in all our experiments. Thus CG represents the difference between the patch n and its neighbouring patch (left, up, right or bottom patch). Of course, here we considered the fact that border patches will not have all four neighbouring patches, and we set these differences to zero. By adding the latter loss term in equation (4.9), we are forcing the local patches to change smoothly. Hence we can alleviate the influence of a noisy U. In Table 4.1, we show a Peak Signal to Noise Ratio

[1] we assumed the reflectance dimension to be s; we did not add any constraint on illuminants.

(PSNR) comparison between our method and method [53]. The PSNR is computed from the separated reflectance image and the ground truth original reflectance image.

Table 4.1: 32-by-32 patches' illumination and reflectance separation performance comparison between our method with [53], here is the reflectance estimation error

Experiments	PSNR	RMS
Original[53]	24.832	0.0573
Improved$^{\lambda=0.05}$	26.476	0.0474

4.3 Local Illuminant Search

The main idea of our method is to divide images into patches, compute local estimations for each patch, and then unify patches into different illuminated regions using CRF. The CRF incorporates spatial information. In this way, we deal with shadow and non-shadow regions separately. Then finally, we run again the local estimation method on regions to get a global estimation for different regions. This pipeline allows us to incorporate spatial information into the segmentation for different illuminated regions.

In order to test our algorithm, we synthesized test images by multiplying reflectance image with daylight illumination. The reflectance image data comes from dataset [51], which contains 22 multispectral reflectance images. We chose the bands from 420nm to 700nm and interpolated at an interval of 5nm.

For the illumination, we synthesized spectra using Judd's spectral power distribution (SPD) function $S(\lambda) = S_0(\lambda) + M_1 S_1(\lambda) + M_2 S_2(\lambda)$ to generate 56 daylight illuminations ranging from colour temperature 4000k to 25000k (4000:100:8500, 9000:500:10000, 11000:1000:15000, 20000, 25000).[2] All daylight bands are the same for the multispectral reflectance image data, again ranging from 420nm to 700 with a interval of 5nm. There are in total 57 bands for each spectrum.

Crucially, as our method is aimed at estimating multiple illuminations in one image, we illuminated the image using two different illuminations, as depicted in Fig. (5.2b).

The first consideration for our method is to divide an image into patches. There are many options to partition images for this purpose. The most straightforward method is to uniformly divide each image into fixed-size patches. Another method is to use the superpixels segmentation method, which tends to segment at object boundaries and significant illumination changes. According to [19, 41, 4], these two methods result in almost the same accuracy much of the time. Hence, here we chose to use the uniform division method. One other reason for this choice is that the local estimation method [53] performs better when

[2]This will make the illuminants to be in 3 dimension space.

there are more colour variations. And uniform dividing provides more diverse colour in each patch and we expect this to persist upon segmentation.

After dividing image into patches, we used the method described in sec.4.2 to estimate illuminations for each patch. Here we add a small modification to this method. Instead of first factorizing the observed image using eq. (4.5), and then solving for illumination L using eq. (4.7), and finally optimizing illumination L and reflectance R via eq. (4.9). Here we simply loop through the possible illuminations from 4000k to 25000k, and find C for each illumination that will minimize eq. (4.9). The illumination that has the minimum loss for eq. (4.9) is chosen as the estimated illumination for that patch. Fig. (5.2d) is the local estimation result for Fig. (5.2a), and we are showing the illuminations' indexes in Fig. (5.2d).

4.4 Different Illuminant Regions Segmentation via CRF

As can be seen from Fig. (5.2d), there seems to be a pattern to partition left and right part. While if simply use a threshold (e.g. 30 in our experiment) to classify the Fig. (5.2d) illumination map, we will obtain the result in Fig. (5.2e). Though Fig. (5.2d) looks like it contains two parts, the partition map is very coarse. The reason in this scene is that if we only use the threshold to control partition then actually we only use the local information for this task. And as we already seen, the local estimation could be very noisy when a patch contains little colour information. Next, we consider using global spatial information for our task.

In addition to assigning each patch label (illumination 1, illumination 2) based on the estimated illumination, the labels should vary smoothly in the same illuminated region, while changes should appear at boundaries for different illuminated regions. The fact that a patch's label depends on the labels of its neighbouring patches allows us to model this optimizing problem as a CRF. The CRF based segmentation model is defined by meaningful spatial relationship between patches. We classify into regions and then re-run the optimization algorithm on different regions again.

Here, the CRF model formulation is similar to other segmentation tasks. Here, we only consider the first order random field, which means only the neighbouring edges between patches are considered. Again, we denotes the multispectral image patches set as P, the edges set as N, and each patch takes a label l_p from $L = \{1, 2\}$, with $\{1, 2\}$ denoting illumination 1 and illumination 2.

Now we use the graph cut model in the Undirected Graphical Models (UGM) toolbox [47] to formulate our CRF problem, and solve for the best labels for each patch. In order to fit our model into UGM toolbox formulation, we wrote the equation in the following form:

$$Potential(l) = \sum_{p \in P} U_p(l_p) + \sum_{(p,q) \in N} B(l_p, l_q), \tag{4.10}$$

where U_p denotes the unary potentials to assign patch p to illumination 1 or illumination 2; larger values of U denote higher confidence that p belongs to that illumination. And $B(l_p, l_q)$ denotes the binary potentials for assigning patch p and q to individual illumination (they could be the same or not). This binary potential represents the confidence with which we believe p and q should be assigned the same label or not.

In our method, we did not see a clear cue to indicate the likelihood for assigning each patch to illumination 1 or 2. So we assign equal potentials to $U_p = \exp^{w_{constant}}$ ($w_{constant} = 0.015$ in our experiment), and therefore we actually do not use the first term in the eq. (4.10).

Now the task is to find the graph cut model for illumination 1 and illumination 2 that maximizes the potential in eq. (4.10). The binary potential function $B(l_p, l_q)$ is defined as:

$$B(l_p, l_q) = \begin{cases} 1 & \text{if } l_p \neq l_q \\ \exp^{w_1 + w_2 * (|Idx_p - Idx_q|/K)} & \text{if } l_p \equiv l_q \end{cases} \tag{4.11}$$

where $K = \max_{\{p,q\} \in N} (|Idx_p - Idx_q|)$ and Idx_p, Idx_q are the estimated illumination indexes from method sec. (4.3). Intuitively, when p and q are illuminated in the same region, B should be larger when assigning $l_p \equiv l_q$ than for $l_p \neq l_q$. In general, if neighbouring patches' local estimation results are similar, then they would have high probability to be illuminated in the same region, and thus we should assign high potentials to $B(l_p, l_q)$ for labeling them the same, and vice versa.

Throughout our experiments, we chose $w_1 = 2.8$ and $w_2 = -1.9$. And this model satisfied our analysis above very well, since when Idx_p and Idx_q are similar, then the potential for B is larger when their labels are the same. Also when Idx_p and Idx_q are different, B is larger when their labels are different.

Finally finding the best labels is finding maximum potentials for eq. (4.10) in this model. Here, the reader is referred the the toolbox [47] for more details about UGM modelling for the graph-cut algorithm. Using the graph-cut decoding algorithm from [47], we segmented the image into different illuminated regions, as shown in Fig. (5.2f).

After obtaining different illuminated regions, we re-run the illumination and reflectance separation algorithm on each different region to get the final result.

Chapter 5

Experiments

5.1 RGB Domain Experiments

As in [20], we used image color and color-gradient information to calculate moments. However, different from [20], we applied our proposed p-Norm and Geo-Mean moments method to color-image and -edge fields. Here we use *p-Norm(color)* to denote using a p-Norm moments calculation for the original image, and use *p-Norm(edge)* to denote applying a p-Norm moments calculation to the gradient-pair image. Similarly, *Geo-Mean(color)* and *Geo-Mean(edge)* denote applying a Geo-Mean moments calculation to color image and color-image edges respectively.

We evaluated our proposed approach on four standard color constancy datasets of real images. For all datasets, we ran 10 runs of three-fold cross validation to train and test our approach. The first dataset used is the re-processed "Gehler" color constancy dataset (described by Lynch et al. in [38]), and denoted here as the Lynch-ColorChecker dataset. This is a re-processed version of an original dataset due to Gehler et al. [42]. The Lynch-ColorChecker dataset contains 482 of these images, which are those taken by a Canon 5D SLR camera; the ground truth illumination is measured from a Macbeth ColorChecker placed in the scene (the Macbeth ColorChecker must be masked off during testing).

In order to compare with other state-of-art algorithms, we also evaluate our methods on the Shi and Funt [49] reprocessed version of the Gehler dataset, which has been commonly used. Two versions of the Shi-Funt reprocessed Gehler dataset have appeared, starting with the 568 images in [49]. These consist of 86 images from a Canon 1D camera and 482 images taken by Canon 5D camera. We use the widely-used original dataset of all 568 images taken as a whole. We refer to this data as the Shi-Funt-Gehler dataset.

Table 5.1: Performance on the reprocessed [38] Color Checker Dataset [42]. Here we use angular error [22] to represent the performance for each method. Asterisk denotes our reproduced result of method [20]

Method	Mean	Median	Tri-Mean	Min	95%
3 Moments*	4.30	3.26	3.57	0.20	10.7
9 Moments*	3.11	2.33	2.49	0.11	8.23
19 Moments*	2.99	2.14	2.29	0.10	8.27
3 Edges*	3.19	2.19	2.44	0.09	8.62
9 Edges*	3.01	**2.03**	2.25	0.12	7.98
19 Edges*	3.20	2.10	2.32	0.10	8.52
3 Geo(color)	3.45	2.74	2.90	0.15	8.63
9 Geo(color)	2.69	2.03	2.22	0.11	6.79
19 Geo(color)	2.58	**1.84**	2.03	0.13	6.85
3 p-Norm(color)$^{p=0.25}$	3.41	2.72	2.87	0.14	8.58
9 p-Norm(color)$^{p=0.25}$	2.59	1.84	2.04	0.10	7.11
19 p-Norm(color)$^{p=0.25}$	2.51	**1.75**	1.95	0.10	6.86
3 Geo(edge)	3.49	2.58	2.83	0.11	9.45
9 Geo(edge)	2.92	2.02	2.21	0.08	7.85
19 Geo(edge)	2.70	**1.88**	2.06	0.14	7.34
3 p-Norm(edge)$^{p=0.25}$	3.33	2.48	2.70	0.13	9.03
9 p-Norm(edge)$^{p=0.25}$	2.76	2.07	2.22	0.10	7.45
19 p-Norm(edge)$^{p=0.25}$	2.87	2.01	2.19	0.10	7.73

Another widely used dataset is the "GrayBall" dataset, so called because the ground truth illumination is measured from an inset physical matte-surface ball in a set of videos. Here we used the 10 images per clip (150 images) compiled by Van der Weijer et al. [28].[1]

The last dataset used is the "SFU Object Dataset" [6], consisting of 321 images of 31 objects under up to 11 different illumination conditions. For all data we carried out the same cross validation process.

In Figure 5.1 we visualize color difference between color-corrected images and ground truth. We display images from [49, 38, 6]. The first column is entirely black since it is error from the ground-truth image: darker indicates smaller color difference.

We can form an *approximated* CIELAB ΔE error image by taking the original image to a standard color space by multiplying by the inverse of a diagonal 3×3 matrix consisting of the ground-truth chromaticity; and similarly we can divide the same image by the approximate, recovered illuminant chromaticity. Then assuming the images are in the standard sRGB color space, first we remove gamma-correction if indeed images were gamma corrected: now we are in linear sRGB color space. Then, in the sRGB standard, a 3×3 matrix transform is defined from linear sRGB to CIE tristimulus XYZ color space. Setting the whitepoint to D65, these triples can then be taken to CIELAB coordinates by the standard

[1]Grayball dataset was taken using NTSC, the data is assumed to be sRGB image. Other datasets are supplied in linear form with no gamma correction applied.

Table 5.2: Performance on the reprocessed [49] Color Checker Dataset [42]. Here we use angular error [22] to represent the performance for each method. Asterisk denotes our reproduced result of method [20]

Method	Mean	Median	Trimean	Min	95%
GrayWorld[21]	6.4	6.3			11.3
Shades of Gray[21]	4.9	4			11.9
GrayEdge[28]	5.1	4.4			11
Gamut Mapping[5]	4.2	2.3			14.1
Spatio-Spectral Statistics[13]	3.4	2.6			9.52
Natural Image Statistics[5]	4.2	3.1			11.7
Exemplar-Based[31]	3.1	2.3			
3 Moments[20]	4.0	3.3			8.9
9 Moments[20]	3.6	2.8			9.1
19 Moments[20]	3.5	2.6			8.6
3 Edges[20]	3.0	2.2			7.2
9 Edges[20]	2.9	2.1			7.1
19 Edges[20]	2.8	2.0			6.9
Moments*	4.48	3.76	4.02	0.19	10.0
9 Moments*	3.81	3.02	3.18	0.17	9.66
19 Moments*	3.61	2.73	2.91	0.13	9.67
3 Edges*	2.97	2.35	2.49	0.09	7.72
9 Edges*	3.32	2.33	2.50	0.12	8.60
19 Edges*	3.35	**2.15**	2.37	0.10	8.86
3 Geo(color)	4.63	3.96	4.14	0.26	10.4
9 Geo(color)	3.49	2.62	2.78	0.16	9.32
19 Geo(color)	3.41	**2.25**	2.46	0.09	9.77
3 p-Norm(color)$^{p=0.25}$	4.60	3.96	4.14	0.20	10.2
9 p-Norm(color)$^{p=0.25}$	3.42	2.59	2.80	0.15	9.26
19 p-Norm(color)$^{p=0.25}$	3.54	2.34	2.58	0.10	10.1
3 Geo(edge)	3.41	2.88	2.96	0.11	8.57
9 Geo(edge)	3.19	2.24	2.48	0.14	8.60
19 Geo(edge)	2.8	**1.95**	2.12	0.13	7.82
3 p-Norm(edge)$^{p=0.25}$	3.08	2.54	2.64	0.12	7.54
9 p-Norm(edge)$^{p=0.25}$	2.78	2.07	2.25	0.12	7.18
19 p-Norm(edge)$^{p=0.25}$	2.79	**2.01**	2.19	0.12	7.62

nonlinear transform. These steps take us to images that are approximately in a perceptually uniform color space, where $\Delta E_{a^*,b^*}$ forms a sensible image-sized indicator of perceptual color difference. As can be seen in the figure, our proposed four moments have smaller error than [20] on the datasets [49, 38]. And on [6], our p-Norm moments are always the best.

Our results appear in Tables 5.1, 5.2, 5.3 and 5.4 for the 4 datasets. As well as citing results for past papers, we also re-implemented the Corrected-Moments method [20] so as to run tests on all the datasets (shown with an asterisk). Our re-implementation found similar

Table 5.3: Performance on the GrayBall dataset, followed [28] pre-processing steps. Asterisk denotes our reproduced result of method [20]

Method	Mean	Median	Tri-Mean	Min	95%
GrayWorld[28]		7.3			
GrayEdge [28]		4.1			
Constrained Minkowski[21]		3.81			
3 Edges[20]		3.8			
9 Edges [20]		3.3			
3 Moments*	4.82	4.07	4.28	0.25	11.4
9 Moments*	3.83	**2.94**	3.17	0.23	10.0
19 Moments*	4.49	3.26	3.45	0.36	12.8
3 Edges*	5.03	4.21	4.32	0.28	13.4
9 Edges*	4.64	3.38	3.68	0.26	13.5
19 Edges*	5.70	3.79	4.15	0.40	16.7
3 Geo(color)	4.77	3.94	4.18	0.22	11.6
9 Geo(color)	4.11	3.09	3.36	0.37	11.1
19 Geo(color)	4.73	3.19	3.45	0.40	12.9
3 p-Norm(color)$^{p=0.5}$	4.88	4.18	4.36	0.28	11.3
9 p-Norm(color)$^{p=0.5}$	3.84	**2.89**	3.12	0.23	10.3
19 p-Norm(color)$^{p=0.5}$	4.68	3.19	3.44	0.37	13.6
3 Geo(edge)	5.96	4.97	5.24	0.44	14.2
9 Geo(edge)	5.55	3.91	4.21	0.27	14.8
19 Geo(edge)	5.86	3.91	4.32	0.43	17.9
3 p-Norm(edge)$^{p=1}$	5.03	4.21	4.32	0.28	13.4
9 p-Norm(edge)$^{p=1}$	4.64	3.38	3.68	0.26	13.5
19 p-Norm(edge)$^{p=1}$	5.70	3.79	4.15	0.40	16.7

results to the reported results in [20], with slight differences due to different parameters for image-size reduction and smoothing.

For optimized parameters, we found that our proposed methods always perform better than our re-implementation of Corrected-Moments, simply by extending the the definition for calculating moments. We speculate that any values poorer than reported in [20] are most likely due to the image shrinking and smoothing processes. Our proposed methods always outperform [20] on all our tested results when we optimize the image pre-processing steps.

5.2 Multispectral Domain Experiments

We use the CRF model to partition the image into two differently illuminated regions. Once the illumination region is partitioned, we again run our illumination and reflectance separation method described in sec. 4.3 on each region to find the best illumination candidate. As we have solved for the illumination for each region, the reflectance can then be solved for

Table 5.4: Performance on the SFU Object dataset[6]. Asterisk denotes our reproduced result of method [20]

Method	Mean	Median	Tri-Mean	Min	95%
GrayWorld	9.8	7.0			
GreyEdge	5.6	3.2			
Gamut Mapping	3.6	2.1			
3 Edges[20]	4.1	3.6			
9 Edges[20]	2.6	**2.0**			
3 Moments*	7.54	5.84	6.10	0.42	19.1
9 Moments*	4.30	3.17	3.48	0.27	11.8
19 Moments*	2.86	2.31	2.43	0.16	7.22
3 Edges*	4.37	3.62	3.76	0.19	11.0
9 Edges*	3.68	**3.00**	3.09	0.24	9.76
19 Edges*	3.58	2.71	2.90	0.23	10.0
3 Geo(color)	7.05	5.55	5.99	0.39	17.4
9 Geo(color)	6.51	4.97	5.35	0.33	16.5
19 Geo(color)	5.27	3.89	4.26	0.33	13.4
3 p-Norm(color)$^{p=2.75}$	5.63	4.40	4.64	0.34	14.9
9 p-Norm(color)$^{p=2.75}$	3.50	2.52	2.75	0.14	10.1
19 p-Norm(color)$^{p=2.75}$	2.92	**1.94**	2.17	0.15	8.71
3 Geo(edge)	5.48	4.48	4.69	0.32	13.9
9 Geo(edge)	4.70	3.99	4.20	0.25	10.9
19 Geo(edge)	4.22	3.33	3.53	0.27	10.9
3 p-Norm(edge)$^{p=2.75}$	4.33	3.50	3.61	0.21	11.9
9 p-Norm(edge)$^{p=2.75}$	3.36	2.49	2.60	0.25	10.0
19 p-Norm(edge)$^{p=2.75}$	3.06	**2.25**	2.47	0.22	7.59

by using the observed image dividing the illumination. In Fig. 5.2, we show the separation result of [53] and of our method.

As can be seen from Fig. 5.2, method [53] almost get the illumination totally wrong. We firstly expect the result would be the average of two illuminations, while as the result is only trying minimizing eq. 4.7, there is no cue for how to balance between these two illuminations. And from Fig. 5.3, we can see the recovered reflectance differs from the original reflectance image, as well as there are two obvious region in the image (which supposed to be uniform like the original reflectance image).

For our method, since we assumed there are two illuminations in the scene, then we can see the recovered two illuminations in Fig. (5.3e) fits the true illumination quite well. And more surprisingly, the recovered reflectance image in Fig. (5.3c) looks quite uniform and close to the original reflectance image. Of course, we note that the wrongly segmented patches in Fig. (5.2f) are the noisy part in our final recovered reflectance image.

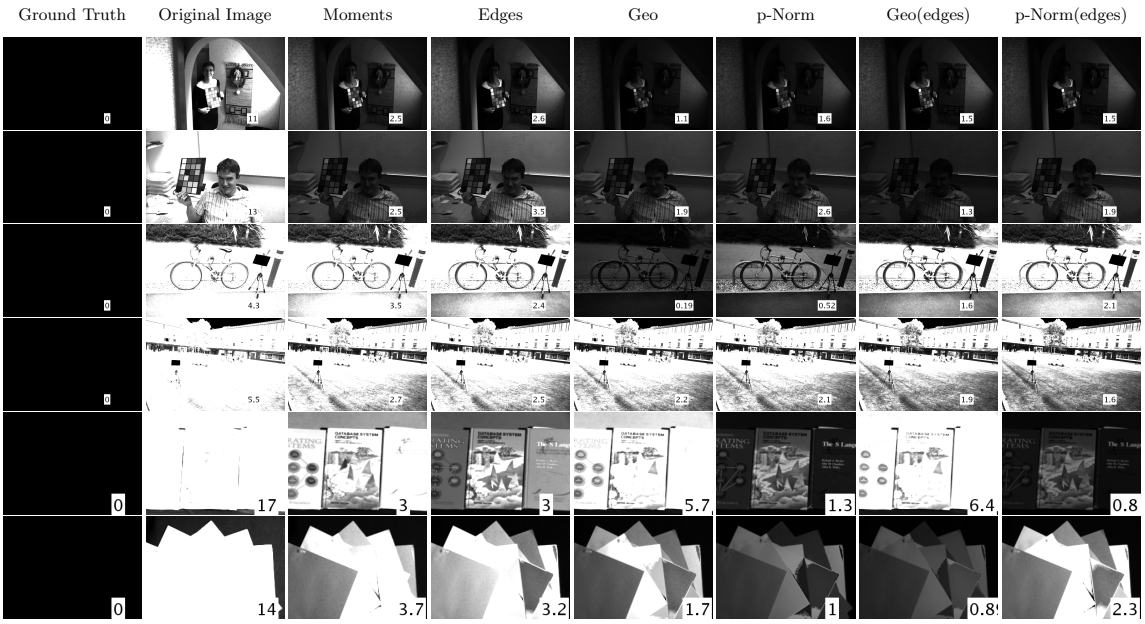
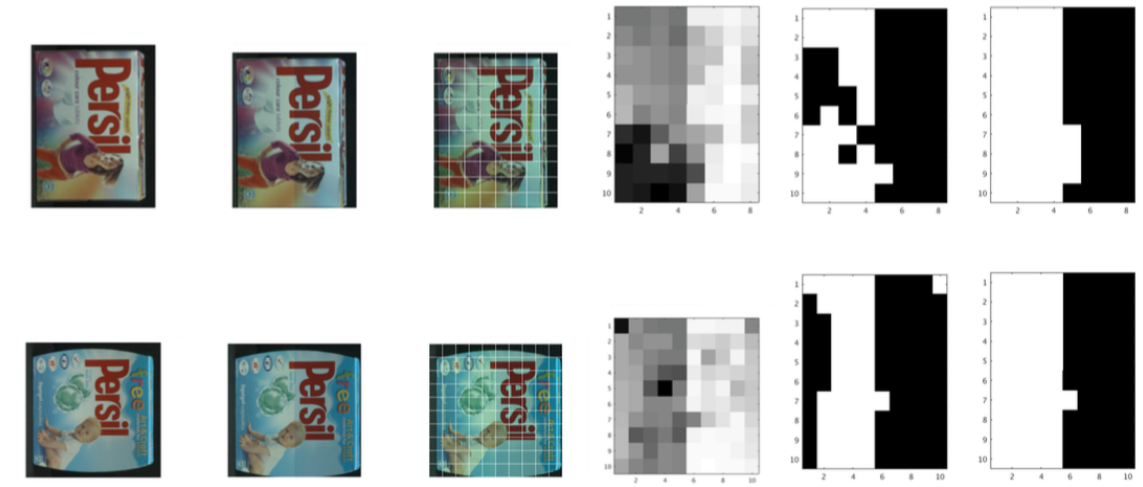
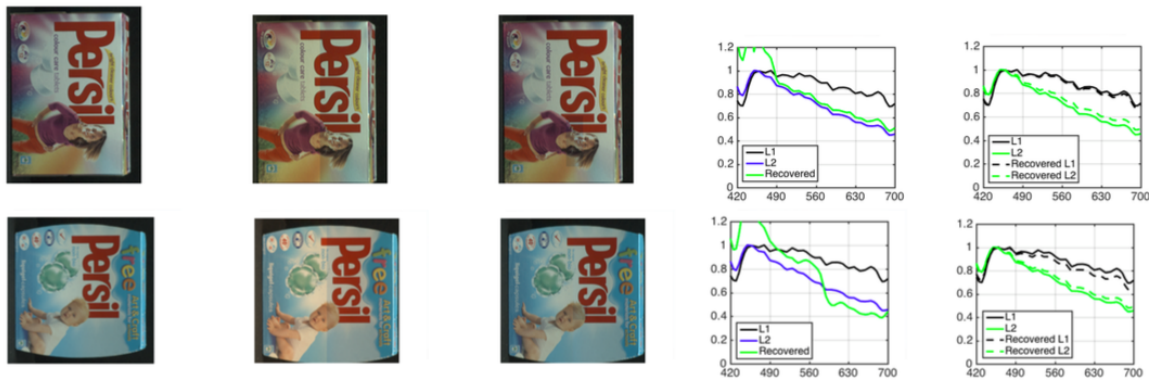


Figure 5.1: Approximated CIELAB error. We first estimated the illuminants for the image, then compute the image reflectance by diving the estimated illuminants. For the reflectance, we first transform it into CIELAB color domain and then calculate the error with the ground-truth. The Moments and Edges columns are from [20]. The last 4 columns are our results.



(a) Reflectance Image (b) 2 illuminations (c) Uniformly divided (d) Local estimation (e) Simple classify (f) Graph cut result

Figure 5.2: All the displaying image are hyper-spectral images, we convert them to RGB image for better display. (a) is the original reflectance image, (b) is the reflectance image illuminated by 4000K daylight on the left half and 8000K daylight on the right half, (c) is the way we divide image into patches, each patch size is 32-by-32, (d) is showing the illumination's index ranging from 1 to 56, (e) is the binary classification result when use index 30 as threshold to segment the image, (f) is the segmented different illuminated regions using graph cut based on local estimation



(a) Reflectance Image (b) [17] recovered (c) Ours recovered (d) [17] recovered (e) Ours recovered

Figure 5.3: (a) is the original reflectance image, (b) is the recovered reflectance image based on method [17] which assume one illumination in the scene, (c) is the recovered reflectance using our method, (d) is recovered illumination using method [17], (e) is recovered illumination using our method

Chapter 6

Conclusion and Future Work

Our contribution can be summarized into two parts. For RGB illumination estimation, we extended the corrected moments idea into several new moments domains. Our ideas originated from success of statistic method for color constancy problem, with the goal to finetune these neat methods based on the corrected moments idea. We show that these strategies can drive down the median and maximum error of illumination estimates.

For the multispectral image, from the experiments given, we have demonstrated that dividing the image into patches helps for multi-illumination estimation, and using a CRF model greatly alleviates the noise introduced by local estimation. Note that our model is not restricted to two illuminations conditions. In eq. (4.10), the labels state could be larger than 2. By simply changing the potential functions eq. (4.11) for more states, our model could handle more illumination conditions. However, of course this extension would definitely introduce more noise into the segmentations.

For future work, we could generate better moments that represent well for the illumination in that scene, or we can even design some un-supervised learned feature to aid the corrected moments method. Also we can design some deep learning network to do illumination estimation that follows our observations in this work. Also we could apply CRF model into multiple RGB illuminants estimation as we performed for multispectral images. For multispectral data, large dataset is needed. Since right now, there is no multiple illuminated multispectral dataset, our work is only based on synthesized image. Also since the multispectral image always contain a lot of data, there is little works using learning based algorithm. As learning based algorithm, especially deep learning has demonstrated big success in other computer vision domain, it might be a future work to use learning based algorithm for multispectral illumination estimation.

Bibliography

[1] Vivek Agarwal, Andrei Gribok, Andreas Koschan, B Abidi, and M Abidi. Illumination chromaticity estimation using linear learning methods. *Journal of Pattern Recognition Research*, 4(1):92–109, 2009.

[2] Vivek Agarwal, Andrei V Gribok, and Mongi A Abidi. Machine learning approach to color constancy. *Neural Networks*, 20(5):559–563, 2007.

[3] Vivek Agarwal, Andrei V Gribok, Andreas Koschan, and Mongi A Abidi. Estimating illumination chromaticity via kernel regression. In *2006 IEEE International Conference on Image Processing*, pages 981–984. IEEE, 2006.

[4] Rui Lu Arjan Gijsenij and Theo Gevers. Color constancy for multiple light sources. *IEEE Transactions on Image Processing*, 21(2):697–707, Feb 2012.

[5] Theo Gevers Arjan Gijsenij and Joost Van De Weijer. Computational color constancy: Survey and experiments. *IEEE Transactions on Image Processing*, 20(9):2475–2489, 2011.

[6] Kobus Barnard, Lindsay Martin, Brian Funt, and Adam Coath. A data set for color research. *Color Research and Application*, 27(3):147–151, 2002.

[7] Jonathan T Barron. Convolutional color constancy. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 379–387, 2015.

[8] Shida Beigpour, Christian Riess, Joost Van De Weijer, and Elli Angelopoulou. Multi-illuminant estimation with conditional random fields. *IEEE Transactions on Image Processing*, 23(1):83–96, 2014.

[9] Simone Bianco, Claudio Cusano, and Raimondo Schettini. Color constancy using cnns. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 81–89, 2015.

[10] Marina G Bloj, Daniel Kersten, and Anya C Hurlbert. Perception of three-dimensional shape influences colour perception through mutual illumination. *Nature*, 402(6764):877–879, 1999.

[11] Gershon Buchsbaum. A spatial processor model for object colour perception. *Journal of the Franklin institute*, 310(1):1–26, 1980.

[12] Vlad C Cardei, Brian Funt, and Kobus Barnard. Estimating the scene illumination chromaticity by using a neural network. *Journal of the Optical Society of America A*, 19(12):2374–2386, 2002.

[13] Ayan Chakrabarti, Keigo Hirakawa, and Todd Zickler. Color constancy with spatio-spectral statistics. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2012.

[14] Po-Rong Chang and Tsung-Hsieh Hsieh. Constrained nonlinear optimization approaches to color-signal separation. *IEEE Transactions on Image Processing*, 4(1):81–94, Jan 1995.

[15] Yangqing Jia Pierre Sermanet Scott Reed Dragomir Anguelov Dumitru Erhan Vincent Vanhoucke Christian Szegedy, Wei Liu and Andrew Rabinovich. Going deeper with convolutions. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9, 2015.

[16] Mark S. Drew and Graham D. Finlayson. Analytic solution for separating spectra into illumination and surface reflectance components. *Journal of the Optical Society of America A*, 24(2):294–303, Feb 2007.

[17] Mark S Drew, Hamid Reza Vaezi Joze, and Graham D Finlayson. The zeta-image, illuminant estimation, and specularity manipulation. *Computer Vision and Image Understanding*, 127:1–13, 2014.

[18] Jonathon Long Evan Shelhamer and Trevor Darrell. Fully convolutional networks for semantic segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 2016.

[19] Pedro F Felzenszwalb and Daniel P Huttenlocher. Efficient graph-based image segmentation. *International Journal of Computer Vision*, 59(2):167–181, 2004.

[20] Graham D Finlayson. Corrected-moment illuminant estimation. In *2013 IEEE International Conference on Computer Vision*, pages 1904–1911. IEEE, 2013.

[21] Graham D Finlayson and Elisabetta Trezzi. Shades of gray and colour constancy. In *Color and Imaging Conference*, pages 37–41. Society for Imaging Science and Technology, 2004.

[22] Graham D Finlayson and Roshanak Zakizadeh. Reproduction angular error: An improved performance metric for illuminant estimation. *perception*, 310(1):1–26, 2014.

[23] Brian Funt and Weihua Xiong. Estimating illumination chromaticity via support vector regression. In *Color and Imaging Conference*, volume 2004, pages 47–52. Society for Imaging Science and Technology, 2004.

[24] Ross Girshick. Fast r-cnn. In *IEEE International Conference on Computer Vision*, pages 1440–1448, 2015.

[25] Michal Mackiewicz Graham D Finlayson and Anya Hurlbert. Root-polynomial colour correction. In *Color and Imaging Conference*, pages 115–119. Society for Imaging Science and Technology, 2011.

[26] Jian Ho, Brian V. Funt, and Mark S. Drew. Separating a color signal into illumination and surface reflectance components: theory and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(10):966–977, Oct 1990.

[27] Evan Shelhamer Jonathan Long and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015.

[28] Theo Gevers Joost van de Weijer and Arjan Gijsenij. Edge-based color constancy. *IEEE Transactions on Image Process*, 16(9):2207–14, Sep 2007.

[29] Theo Gevers Joost Van De Weijer and Arjan Gijsenij. Edge-based color constancy. In *IEEE Transaction on Image Processing*, 2007.

[30] Ross Girshick Joseph Redmon, Santosh Divvala and Ali Farhadi. You only look once: Unified, real-time object detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 779–788, 2016.

[31] Hamid Reza Vaezi Joze and Mark S Drew. Exemplar-based colour constancy and multiple illumination. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Aug 2013.

[32] J. Hallikainen Jussi PS Parkkinen and T. Jaaskelainen. Characteristic spectra of munsell colors. *Journal of the Optical Society of America A*, 6(2):318–322, 1989.

[33] Shaoqing Ren Kaiming He, Xiangyu Zhang and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[34] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[35] Edwin H Land and John J McCann. Lightness and retinex theory. *Journal of the Optical Society of America A*, 61(1):1–11, 1971.

[36] Ze-Nian Li, Mark S Drew, and Jiangchuan Liu. *Fundamentals of Multimedia, 2nd. ed.* Springer, 2014.

[37] Cheng Lu and Mark S Drew. Shadow segmentation and shadow-free chromaticity via markov random fields. In *Color and Imaging Conference*, volume 2005, pages 125–129. Society for Imaging Science and Technology, 2005.

[38] Stuart E Lynch, Mark S Drew, and Graham D Finlayson. Colour constancy from both sides of the shadow edge. In *IEEE International Conference on Computer Vision Workshops*, pages 899–906. IEEE, 2013.

[39] Laurence T Maloney. Evaluation of linear models of surface spectral reflectance with small numbers of parameters. *Journal of the Optical Society of America A*, 3(10):1673–1683, 1986.

[40] Seoung Wug Oh and Seon Joo Kim. Approaching the computational color constancy as a classification problem through deep learning. *Pattern Recognition*, 61:405–416, 2017.

[41] Yuri Boykov Olga Veksler and Paria Mehrani. Superpixels and supervoxels in an energy optimization framework. In *Computer Vision–ECCV 2010*, pages 211–224. Springer, 2010.

[42] Andrew Blake Tom Minka Peter Vincent Gehler, Carsten Rother and Toby Sharp. Bayesian color constancy revisited. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[43] Youngjun Yoo Rajeev Ramanath, Wesley E Snyder and Mark S Drew. Color image processing pipeline. *IEEE Signal Processing Magazine*, 22(1):34–43, 2005.

[44] Ko Nishino Robby T Tan and Katsushi Ikeuchi. Illumination chromaticity estimation using inverse-intensity chromaticity space. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages I–673. IEEE, 2003.

[45] Antonio Robles-Kelly and Cong Phuoc Huynh. *Imaging spectroscopy for scene analysis*. Springer Science & Business Media, 2012.

[46] Trevor Darrell Ross Girshick, Jeff Donahue and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 580–587, 2014.

[47] M. Schmidt. Ugm: A matlab toolbox for probabilistic undirected graphical models, 2007.

[48] Ross Girshick Shaoqing Ren, Kaiming He and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[49] Lilong Shi and Brian Funt. Re-processed version of the Gehler color constancy dataset of 568 images. *Simon Fraser University*, 2010.

[50] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[51] Graham Finalyson Steven Hordley and Peter Morovic. A multi-spectral image database and its application to image rendering across illumination. In *International Conference on Image and Graphics*, pages 394–397, Dec 2004.

[52] Xu De Wang Ning and Li Bing. Edge-based color constancy via support vector regression. *IEEE Transactions on Information and Systems*, 92(11):2279–2282, 2009.

[53] Imari Sato Yinqiang Zheng and Yoichi Sato. Illumination and reflectance spectra separation of a hyperspectral image meets low-rank matrix factorization. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1779–1787, June 2015.

[54] Mark S Drew Ze-Nian Li and Jiangchuan Liu. *Fundamentals of multimedia*. Springer, 2004.

[55] Ninghang Hu Marcel P Lucassen Zhongyu Lou, Theo Gevers. Color constancy by deep learning. 2015.