Extended Corrected-Moments Illumination Estimation

Xiaochuan Chen*[1]*, Mark S. Drew*[1]*, Ze-Nian Li*[1]*, and Graham D. Finlayson*[2]*;* [1] *Simon Fraser University,* {*xca64,mark,li*}*@cs.sfu.ca,
[2] ***University of East Anglia, graham@cmp.uea.ac.uk***

Abstract

A remarkably simple color constancy method was recently developed, based in essence on the Gray-Edge method, i.e., the assumption that the mean of color-gradients in a scene (or colors themselves, in a Gray-World setting) are close to being achromatic. However this new method for illuminant estimation explicitly includes the important notions that (1) we cannot hope to recover illuminant strength, but only chromaticity; and (2) that a polynomial regression from image moment vectors to chromaticity triples should be based not on polynomials but instead on the roots of polynomials, in order to release the regression from absolute units of lighting. In this paper we extend these new image moments in several ways: by replacing the standard expectation value mean used in the moments by a Minkowski p-norm; by going over to a float value for the parameter p and carrying out a nonlinear optimization on this parameter; 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 illumination estimates.

Introduction

Colors in images result from the combination of illumination, surface reflection, and camera sensors plus the effects of the imaging and display pipeline [13]. In general, the human visual system is capable of filtering out the effects of the illumination 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 illumination conditions. The computer science goal in the color constancy task is to estimate the illumination, or at least the chromaticity – color without magnitude. Remarkably, the recent Corrected Moments illumination estimation due to Finlayson [6] does overall best in terms of illumination 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 paper we re-examine Finlayson's Corrected Moments method [6] with a view to simple extensions which we find further improve the illumination 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, thus surpassing 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 illumination estimation. Then we show how to incorporate the Zeta-Image [5] 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.

The paper is organized as follows. In Section **[Related Work]** we discuss related works that form the scaffold for the present work. In Section **[Corrected Moments]** we review the corrected moments approach proposed by [6]. In the Section **[Minkowski Norm and Geometric Mean in Corrected Moments Method]** we propose novel moments to be used in the Corrected-Moments approach, plus a new optimization scheme. We compare results for the proposed moments with results obtained previously by exhaustively considering different estimators applied to 4 standard datasets.

Related Work

Gray-World and Gray-Edge

In experiments and tables of results below, note that we compare results with the best to date, state-of-the-art methods. However, in fact the method in [6] is based on very simple algorithms, so we begin the discussion with these. The simplest illumination estimation algorithm is the Gray-World algorithm [3], which assumes that the average reflectance in a scene is achromatic. Thus the illumination 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 $E(R_k) = \frac{1}{N}\sum_{i=1}^{N} R_k^i$, with N being the number of pixels. Intuitively, Gray-World will obviously fail if a scene is insufficiently colorful. For example, an image of a gold coin that takes up most of the pixels will generate a very poor illumination estimate; and if we move the white point to $R = G = B = 1$, or use a more careful white-point camera balance (see, e.g., [11]) then our image will likely end up containing a coin that looks gray rather than gold.

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 [16]. With this assumption, the illumination 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 illumination color [16]. The expectation value for the kth color channel is then estimated by

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

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

Based on the Gray-World and Gray-Edge approaches, many extensions have been proposed. The Shades-of-Gray algorithm [8] 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 illumination estimation performance [8]. This estimate is as follows:

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

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

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

$$\hat{c}_k = \sqrt[p]{\sum_{i=1}^{N} |\frac{\delta_\sigma^n R_k^i}{\delta_\sigma x^n}|^p + |\frac{\delta_\sigma^n R_k^i}{\delta_\sigma y^n}|^p} \tag{3}$$

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

Ref.[10] presents a useful review of various CC algorithms, some of which are quite complex. In eq. (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[8]. An advantage of all the color constancy approaches based on (3) is that they are of low computational complexity both in space and in time [16]. 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.

Corrected Moments

To date, most recently the most successful descendant of the methods above is that in [6]. The method in [6] derives from Eq. (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, illumination estimation performance bested state-of-art approaches involving a great deal of computation and even feature extraction (see [4, 6, 8, 17, 10, 15]).

In this paper 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 [5]. And thirdly, we also replace the corrected moments method use of gradient data as in Eq. (1) by instead using a Minkowski p-norm. We show that these strategies can drive down the median and maximum error of illumination estimates.

Polynomial Regression

The Corrected-Moments method can be understood as a variant of polynomial regression. Here we briefly recapitulate the method [6], 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. (1) and in Eq. (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 interreflections at the corners, which arguably may help the human visual system disambiguate the situation [2]. 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 [6], 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

$$\min_{A} ||\mathbf{MA} - \mathbf{C}||_2 \tag{4}$$

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

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.(4), 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 [7] 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 [7], this insight was utilized in the color correction problem – mapping RGB to tristimulus values XYZ. In [6] this same insight was further applied to the illumination 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 [6] 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 [6] the author sets up an optimization

$$\min_{K,A} ||\mathbf{KMA} - \mathbf{C}||_2 \tag{5}$$

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 $\mathbf{k} = diag(\mathbf{K})$ [6] (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 pseudoinverse: 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 [6] applies the above algorithm to RGB image data $\mathbf{R}$, or alternatively to gradient data $\nabla\mathbf{R}$, 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 by 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 [6] are two-fold: a reduction in illumination estimate error is observed, and also a reduction in the maximum error – the latter is important because manufacturers desire no perceptible outlier cases.

In [6], the author used RGB moments as well as higher order RGB moments. These moments can be summarized as follows:

$$m_{uvw} = \left[\frac{\sum_{i=1}^{N} R_i^u G_i^v B_i^w}{N} \right]^{1/d} \qquad u+v+w = d; \quad u,v,w \geq 0 \tag{6}$$

Here R,G,B is used as a shorthand for either RGB pixel value themselves or Gaussian first (in [6]) 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 (5) scale with intensity. And this "intensity scaling" property is important for correcting illumination estimation [6]: 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 $\nabla\mathbf{R}$; 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.

Minkowski Norm and Geometric Mean in Corrected Moments Method

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 illumination estimate, we produce moments as follows: for the linear, d=1 moments we use

$$(m_k)_{(d=1)} = (\frac{1}{N} \sum_{i=1}^{n} (R_k^i)^p)^{1/p}, \quad k = 1..3 \tag{7}$$

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* $_{(3+j-1)}C_j$ *= 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

Optimization of p-Norm

In the above, the 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).

Corrected Moments with Geometric Mean

In [5] 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 equation (6), our geometric mean moments may be summarized in the following equation:

$$m_{uvw} = \left[\left(\prod_{i=1}^{N} R_i^u G_i^v B_i^w\right)^{1/N}\right]^{1/d} \quad u+v+w = d \;\; u,v,w \geq 0 \quad (8)$$

Here m_{uvw} now denotes the new moments. When we set $d = 1$, m_{uvw} is simply the illumination color triple. Note that in [5], 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. 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 illumination estimation [6].

Application of Minkowski p-Norm and Geo-Mean in Corrected Moments

As in [6], we used image color and color-gradient information to calculate moments. However, different from [6], we applied our proposed p-Norm and Geo-Mean moments method to color-image and -edge fields, instead of using the mean as expectation value. 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 threefold crossvalidation to train and test our approach. The first dataset used is the reprocessed "Gehler" color constancy dataset (described by Lynch et al. in [12]), and denoted here as the Lynch-ColorChecker dataset. This is a re-processed version of an original dataset due to Gehler et al. [9]. 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 training and testing).

In order to compare with other state-of-art algorithms, we also evaluate our methods on the Shi and Funt [14] 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 [14]. 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.[1] We refer to this data as the Shi-Funt-Gehler dataset.

Another widely used dataset is the "GrayBall" dataset, so called because the ground truth illumination is measured from an inset physical matte-surface gray ball in a set of videos. Here we used the 10 images per clip (150 images) compiled by Van der Weijer et al. [16].

[1]The original Shi-Funt reprocessed Gehler dataset was removed on Sept. 16, 2013 and replaced: here we use the original data in order to correctly compare with other methods that used the original dataset.

The last dataset used is the "SFU Object Dataset" [1], 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 1 we visualize color difference between color-corrected images and ground truth. We display images from [14, 12, 1]. 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 chromaticiy; 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 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 [6] on the datasets [14, 12]. And on [1], our p-Norm moments are always the best.

Our results appear in Tables 1, 2, 3, and 4 for the 4 datasets. As well as citing results for past papers, we also re-implemented the Corrected-Moments method [6] so as to run tests on all the datasets (shown with an asterisk). Our re-implementation found similar results to the reported results in [6], 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 definition for calculating moments. We speculate that any values poorer than reported in [6] are most likely due to the image shrinking and smoothing processes. Our proposed methods always outperform [6] on all our tested results when we optimize the image preprocessing steps.

References

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

[2] 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.

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

[4] Ayan Chakrabarti, Keigo Hirakawa, and Todd Zickler. Color constancy with spatio-spectral statistics. *IEEE PAMI*, 2012.

[5] 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.

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

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

[8] 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.

[9] P. Gehler, C. Rother, A. Blake, T. Minka, and T. Sharp. Bayesian color constancy revisited. In *CVPR'08: Computer Vision and Patt. Rec.*, 2008.

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

[11] Z.-N. Li, M.S. Drew, and J. Liu. *Fundamentals of Multimedia, 2nd. ed.* Springer, 2014.

[12] 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 (ICCVW)*, pages 899–906. IEEE, 2013.

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

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

[15] Hamid Reza Vaezi Joze and Mark S Drew. Exemplar-based colour constancy and multiple illumination. *IEEE Trans Pattern Anal Mach Intell*, Aug 2013.

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

[17] J. V. D. Weijer, T. Gevers, and A. Gijsenij. Edge-based color constancy. In *IEEE Trans. on Im. Proc.*, 2007.

Author Biography

Mark S. Drew is a Professor in the School of Computing Science at Simon Fraser University in Vancouver, Canada. His background education is in Engineering Science, Mathematics, and Physics. His interests lie in the fields of image processing, color, computer vision, computer graphics, multimedia, and visualization. He has published over 160 refereed papers and is the holder of 9 patents and applications in computer vision, image processing, and image reconstruction and accenting.

Table 1: Shi and Funt Original Dataset (pre-16Sept2013)

Method	Mean	Median	Trimean	Min	95%
GrayWorld[8]	6.40	6.30			11.30
Shades of Gray[8]	4.90	4.00			11.90
GrayEdge[16]	5.10	4.40			11.00
Gamut Mapping[10]	4.20	2.30			14.10
Spatio-Spectral Statistics[4]	3.40	2.60			9.52
Natural Image Statistics[10]	4.20	3.10			11.70
Exemplar-Based[15]	3.10	2.30			
3 Moments[6]	4.00	3.30			8.90
9 Moments[6]	3.60	2.80			9.10
19 Moments[6]	3.50	2.60			8.60
3 Edges[6]	3.00	2.20			7.20
9 Edges[6]	2.90	2.10			7.10
19 Edges[6]	2.80	2.00			6.90
3 Moments*	4.49	3.77	4.03	0.20	10.04
9 Moments*	3.82	3.02	3.18	0.17	9.67
19 Moments*	3.62	2.73	2.91	0.13	9.68
3 Edges*	2.98	2.35	2.50	0.10	7.73
9 Edges*	3.32	2.34	2.50	0.11	8.60
19 Edges*	3.35	2.1532	2.37	0.10	8.87
3 Geo(color)	4.63	3.97	4.15	0.25	10.40
9 Geo(color)	3.50	2.62	2.79	0.15	9.32
19 Geo(color)	3.42	2.2543	2.47	0.09	9.78
3 p-Norm(color)$^{p=0.25}$	4.60	3.96	4.14	0.20	10.22
9 p-Norm(color)$^{p=0.25}$	3.43	2.59	2.81	0.14	9.27
19 p-Norm(color)$^{p=0.25}$	3.54	2.34	2.58	0.09	10.17
3 Geo(edge)	3.41	2.89	2.96	0.11	8.58
9 Geo(edge)	3.19	2.25	2.48	0.14	8.60
19 Geo(edge)	2.80	1.9531	2.12	0.14	7.82
3 p-Norm(edge)$^{p=0.25}$	3.09	2.54	2.65	0.13	7.54
9 p-Norm(edge)$^{p=0.25}$	2.78	2.08	2.25	0.12	7.19
19 p-Norm(edge)$^{p=0.25}$	2.80	2.0117	2.19	0.11	7.63

Table 2: Lynch-ColorChecker Dataset

Method	Mean	Median	Tri-Mean	Min	95%
3 Moments*	4.31	3.26	3.58	0.19	10.73
9 Moments*	3.12	2.34	2.50	0.10	8.23
19 Moments*	2.99	2.14	2.30	0.09	8.27
3 Edges*	3.19	2.20	2.44	0.08	8.63
9 Edges*	3.02	2.0346	2.25	0.13	7.99
19 Edges*	3.21	2.11	2.33	0.11	8.52
3 Geo(color)	3.45	2.74	2.91	0.16	8.63
9 Geo(color)	2.69	2.03	2.23	0.11	6.80
19 Geo(color)	2.58	1.8364	2.03	0.13	6.86
3 p-Norm(color)$^{p=0.25}$	3.42	2.72	2.88	0.14	8.58
9 p-Norm(color)$^{p=0.25}$	2.60	1.85	2.05	0.09	7.11
19 p-Norm(color)$^{p=0.25}$	2.51	1.7476	1.96	0.10	6.86
3 Geo(edge)	3.50	2.58	2.83	0.11	9.45
9 Geo(edge)	2.92	2.02	2.22	0.08	7.86
19 Geo(edge)	2.71	1.8773	2.06	0.14	7.34
3 p-Norm(edge)$^{p=0.25}$	3.34	2.49	2.70	0.14	9.04
9 p-Norm(edge)$^{p=0.25}$	2.76	2.08	2.22	0.10	7.45
19 p-Norm(edge)$^{p=0.25}$	2.88	2.01	2.19	0.10	7.74

Table 3: GrayBall Dataset

Method	Mean	Median	Tri-Mean	Min	95%
GrayWorld[16]		7.30			
GrayEdge [16]		4.10			
Constrained Minkowski[8]		3.81			
3 Edges[6]		3.80			
9 Edges [6]		3.30			
3 Moments*	4.82	4.08	4.28	0.24	11.49
9 Moments*	3.84	2.9361	3.17	0.24	10.06
19 Moments*	4.49	3.26	3.45	0.36	12.83
3 Edges*	5.04	4.21	4.32	0.28	13.41
9 Edges*	4.65	3.38	3.69	0.26	13.60
19 Edges*	5.70	3.80	4.16	0.40	16.70
3 Geo(color)	4.78	3.94	4.19	0.23	11.62
9 Geo(color)	4.12	3.09	3.37	0.37	11.16
19 Geo(color)	4.74	3.19	3.45	0.40	12.97
3 p-Norm(color)$^{p=0.5}$	4.88	4.19	4.37	0.28	11.31
9 p-Norm(color)$^{p=0.5}$	3.85	2.8864	3.12	0.23	10.34
19 p-Norm(color)$^{p=0.5}$	4.69	3.20	3.45	0.36	13.66
3 Geo(edge)	5.96	4.98	5.25	0.45	14.27
9 Geo(edge)	5.56	3.91	4.21	0.27	14.88
19 Geo(edge)	5.86	3.91	4.33	0.44	17.97
3 p-Norm(edge)$^{p=1}$	5.04	4.21	4.32	0.28	13.41
9 p-Norm(edge)$^{p=1}$	4.65	3.38	3.69	0.26	13.60
19 p-Norm(edge)$^{p=1}$	5.70	3.80	4.16	0.40	16.70

Table 4: SFU Object Dataset

Method	Mean	Median	Tri-Mean	Min	95%
GrayWorld	9.80	7.00			
GreyEdge	5.60	3.20			
Gamut Mapping	3.60	2.10			
3 Edges[6]	4.10	3.60			
9 Edges[6]	2.60	2.0			
3 Moments*	7.54	5.85	6.10	0.42	19.11
9 Moments*	4.31	3.17	3.48	0.27	11.83
19 Moments*	2.87	2.32	2.43	0.15	7.23
3 Edges*	4.37	3.62	3.77	0.20	11.06
9 Edges*	3.69	2.9974	3.09	0.24	9.76
19 Edges*	3.58	2.71	2.90	0.23	10.00
3 Geo(color)	7.05	5.55	6.00	0.39	17.49
9 Geo(color)	6.52	4.98	5.35	0.34	16.57
19 Geo(color)	5.28	3.89	4.26	0.34	13.41
3 p-Norm(color)$^{p=2.75}$	5.63	4.41	4.64	0.34	14.90
9 p-Norm(color)$^{p=2.75}$	3.51	2.53	2.76	0.14	10.12
19 p-Norm(color)$^{p=2.75}$	2.93	1.9366	2.17	0.16	8.72
3 Geo(edge)	5.49	4.48	4.69	0.33	13.98
9 Geo(edge)	4.70	3.99	4.21	0.26	10.90
19 Geo(edge)	4.22	3.34	3.54	0.27	10.91
3 p-Norm(edge)$^{p=2.75}$	4.33	3.51	3.61	0.21	11.90
9 p-Norm(edge)$^{p=2.75}$	3.37	2.50	2.60	0.24	10.06
19 p-Norm(edge)$^{p=2.75}$	3.06	2.2466	2.48	0.22	7.60

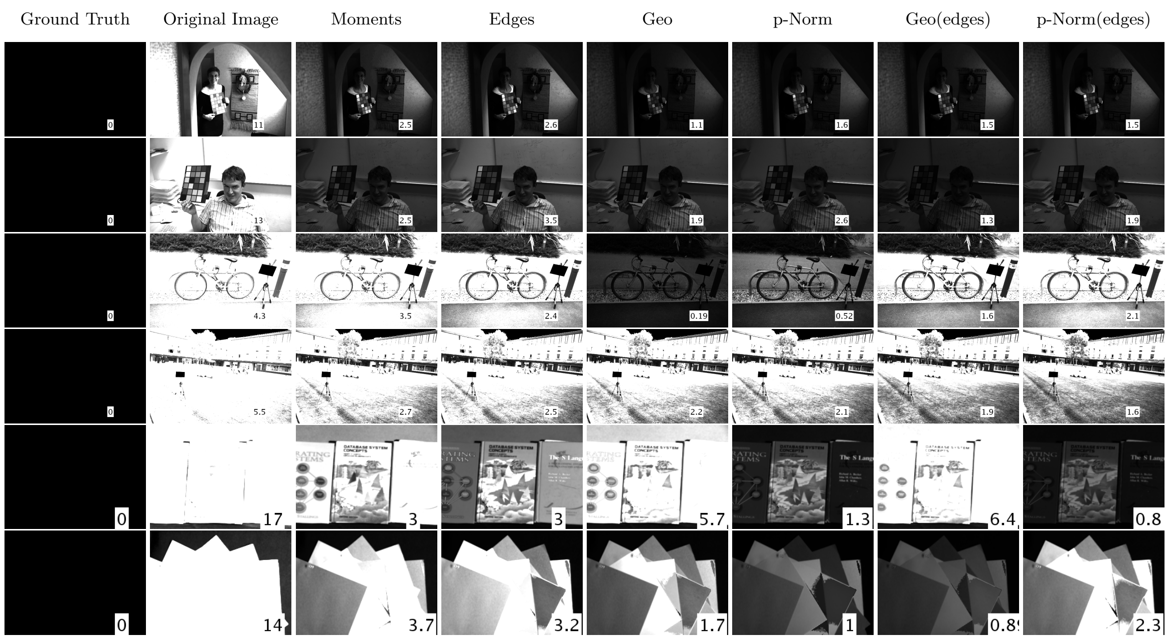

Figure 1. *Approximate CIELAB error.*