Color by Correlation: A Simple, Unifying Framework for Color Constancy

Graham D. Finlayson, Steven D. Hordley, and Paul M. Hubel

Abstract—This paper considers the problem of illuminant estimation: how, given an image of a scene, recorded under an unknown light, we can recover an estimate of that light. Obtaining such an estimate is a central part of solving the color constancy problem—that is of recovering an illuminant independent representation of the reflectances in a scene. Thus, the work presented here will have applications in fields such as color-based object recognition and digital photography, where solving the color constancy problem is important. The work in this paper differs from much previous work in that, rather than attempting to recover a single estimate of the illuminant as many previous authors have done, we instead set out to recover a measure of the likelihood that each of a set of possible illuminants was the scene illuminant. We begin by determining which image colors can occur (and how these colors are distributed) under each of a set of possible lights. We discuss in the paper how, for a given camera, we can obtain this knowledge. We then correlate this information with the colors in a particular image to obtain a measure of the likelihood that each of the possible lights was the scene illuminant. Finally, we use this likelihood information to choose a single light as an estimate of the scene illuminant. Computation is expressed and performed in a generic correlation framework which we develop in this paper. We propose a new probabilistic instantiation of this correlation framework and we show that it delivers very good color constancy on both synthetic and real images. We further show that the proposed framework is rich enough to allow many existing algorithms to be expressed within it: the gray-world and gamut-mapping algorithms are presented in this framework and we also explore the relationship of these algorithms to other probabilistic and neural network approaches to color constancy.

Index Terms—Color constancy, illuminant estimation, correlation matrix.

1 Introduction

An image of a three-dimensional scene depends on a number of factors. First, it depends on the physical properties of the imaged objects, that is on their reflectance properties. But, it also depends on the shape and orientation of these objects and on the position, intensity, and color of the light sources. Finally, it depends on the spectral sampling properties of the imaging device. Various applications require that we be able to disambiguate these different factors to recover the surface reflectance properties of the imaged objects or the spectral power distribution (SPD) of the incident illumination. For example, in a number of applications ranging from machine vision tasks, such as object recognition, to digital photography, it is important that the colors recorded by a device are constant across a change in the scene illumination. As an illustration, consider using the color of an object as a cue in a recognition task. Clearly, if such an approach is to be successful, then this color must be stable across illumination change [28]. Of course, it is not stable since changing the color of the illumination changes the color of the light reflected from an object. Hence, a preliminary step in using color for object recognition must be to remove the effect of illumination color from the object to be recognized. Accounting for the prevailing illumination is important in photography, too; it is known [1], [3] that the human visual system corrects, at least partially, for the prevailing scene illumination; therefore, if a photograph is to be an accurate representation of what the photographer saw, the photograph must be similarly corrected.

Central to solving the color constancy problem is recovering an estimate of the scene illumination and it is that problem which is the focus of this paper. Specifically, we consider how, given an image of a scene taken under an unknown illuminant, we can recover an estimate of that light. We present in this paper, a simple new approach to solving this problem, which requires only that we have some knowledge about the range and distribution of image colors which can be recorded by a camera under a set of possible lights. We discuss later in the paper how such knowledge can be obtained.

The work presented here builds on a range of computational theories previously proposed by other authors [21], [23], [8], [9], [4], [5], [27], [29], [14], [10]. The large number of such theories illustrates that the problem is difficult to solve. Part of the difficulty is due to the fact that the problem is inextricably tied up with other confounding phenomena: We have to account for changes in image intensity and color which are due to the shape of the objects, viewing and illumination geometry, as well as those due to changes in the spectral power distribution of the illuminant and the spectral reflectance properties of the imaged objects. Thus, to simplify the problem, many researchers [21], [23], [14], [8] have considered a simplified two-dimensional world in which all objects are flat, matte, Lambertian surfaces,

- *G.D. Finlayson and S.D. Hordley are with the School of Information Systems, University of East Anglia, Norwich NR4 7TJ, UK. E-mail: {steve, graham}@sys.uea.ac.uk.*
- *P.M. Hubel is with Hewlett-Packard Labs, Hewlett Packard Inc., 1501 Page Mill Road, Palo Alto, CA 94306. E-mail: paul_hubel@hp.com.*

Manuscript received 26 July 1999; revised 25 Sept. 2000; accepted 9 Mar. 2001.
Recommended for acceptance by S. Sarkar.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number 110311.

uniformly illuminated. In this idealized scenario, image formation can be described by the following simple equation:

$$p_k^x = \int_\omega E(\lambda) S^{\mathbf{x}}(\lambda) R_k(\lambda) d\lambda. \quad (1)$$

In this equation, $S^{\mathbf{x}}(\lambda)$ represents the surface reflectance at a point $\mathbf{x}$ in the scene: It defines what fraction of the incident light is reflected on a per wavelength basis. $E(\lambda)$ is the spectral power distribution of the incident illuminant which defines how much power is emitted by the illuminant at each wavelength. $R_k(\lambda)$ is the relative spectral response of the imaging device's $k\text{th}$ sensor, which specifies what proportion of the light incident at the sensor is absorbed at each wavelength. These three terms when multiplied together and the product integrated over the interval ω (the range of wavelengths to which the sensor is sensitive) gives p_k^x: the response of the imaging device's $k\text{th}$ sensor at pixel x. It is clear from (1) that changing either the surface reflectance function or the spectral power distribution of the illuminant will change the values recorded by the imaging device. The task for a color constancy algorithm is to transform the p_k^x so that they become independent of $E(\lambda)$ and, hence, correlate with $S(\lambda)$. Equivalently, the problem can be posed as that of recovering an estimate of $E(\lambda)$ since with this knowledge, it is relatively straightforward [19] to recover an image which is independent of the prevailing illumination.

Even with the simplified model of image formation in (1), the problem remains difficult to solve. To see why, consider a typical imaging device with three classes of sensors: $k = 3$ (it is common to refer to the triplet of sensor responses (p_1^x, p_2^x, p_3^x) as R, G, and B, or simply RGB—because typically sensors measure the long (red), medium (green), and short (blue) wavelengths, respectively. These expressions are used interchangeably throughout this paper). Now, if there are n different surfaces in an image, then we have $3n$ *knowns* from (1). From these equations, we must estimate parameters for the n surfaces and a single illuminant. Surface reflectance functions and illuminant SPDs, as well as sensor response functions, are typically specified by their value at a number (m) of discrete sample points within the visible spectrum. In this case, the image formation equation (1) can be rewritten as:

$$p_k^x = \sum_{i=1}^{i=m} E(\lambda_i) S^x(\lambda_i) R_k(\lambda_i) \Delta\lambda, \quad (2)$$

where the λ_is are the sample points and $\Delta\lambda$ is the width between them. If surfaces and illuminants are each described by m parameters in this way, then we have a total of $m(n+1)$ parameters to solve for. It is clear that the number of knowns $3n$ can never be bigger than the number of unknowns $m(n+1)$ regardless of how many distinct surfaces appear in a scene.

Fortunately, it is often unnecessary to recover the full spectra of lights and surfaces, rather it is sufficient to represent a light by the response of a device to a perfect diffuser viewed under it and, similarly, to represent a surface by the response it induces under some canonical light. Continuing with the case of an imaging device with three classes of sensor, this implies that lights and surfaces are described by three parameters each, so that the total number of parameters to be solved for is $3n + 3$ and the problem is thus further constrained. Nevertheless, the problem is still underconstrained; there are three more unknowns than knowns to solve for.

In this paper, we make one further simplification of the problem: rather than representing lights and surfaces by a 3-vector of sensor responses, $(p_1, p_2, p_3)^t$, we instead represent them in terms of their 2D *chromaticity* vectors, $(c_1, c_2)^t$, calculated from the original sensor response by discarding intensity information. There are many ways in which we might discard intensity information: One common way is to divide two of the sensor responses by the response of the third:

$$c_1^x = \frac{p_1^x}{p_3^x} \quad c_2^x = \frac{p_2^x}{p_3^x}. \quad (3)$$

Ignoring intensity information means that changes in surface color due to geometry or viewing angle, which change only intensity, will not affect our computation and, in addition, we have reduced the problem from a 3D one to a 2D one. Furthermore, we point out that, under the model of image formation described by (1), illumination can only be recovered up to a multiplicative constant.[1] However, even with this simplification, we still have $2(n+1)$ unknowns and $2n$ knowns, so that the problem remains underconstrained.

Many authors [21], [5], [18], [23], [8] have tried to deal with the underconstrained nature of the color constancy problem by making additional assumptions about the world. For example, Land [21] assumes that every image contains a white patch, hence, there are now only $3n$ unknowns and $3n$ equations. Another assumption [5], [18] is that the average reflectance of all surfaces in a scene is achromatic. In this case, the average color of the light leaving the surface will be the color of the incident illumination. Yet, another approach [23], [8] has been to model lights and surfaces using low-dimensional linear models and to develop recovery schemes which exploit the algebraic features of these models. Other authors have tried to exploit features not present in the idealized Mondrian world, such as specularities [22], [27], [29], shadows [11], or mutual illumination [15], to recover information about the scene illuminant. Unfortunately, the assumptions made by all these algorithms are quite often violated in real images so that many of the algorithms work only inside the lab [15], [22], [29] and, while others can work on real images, their performance is still short of good enough color constancy [16].

The fact that the problem is underconstrained implies that, in general, the combination of surfaces and illuminant giving rise to a particular image is not unique. So, setting out to solve for a unique answer (the goal of most algorithms) is perhaps not the best way to proceed. This

1. Within our model of image formation, the light incident at the imaging device is the product of illuminant spectral power and surface reflectance; $E(\lambda)S(\lambda)$. Clearly, the product $sE(\lambda)\frac{S(\lambda)}{s}$ will result in the same incident light for any value of s. Hence, $E(\lambda)$ can only be recovered up to a multiplicative constant.

point of view has only been considered in more recent algorithms. For example, the gamut-mapping algorithms developed by Forsyth [14] and, later, by Finlayson [10] and Finlayson and Hordely [12], do not, in the first instance, attempt to find a unique solution to the problem; rather, the set of all possible solutions are found and, from this set, the *best* solution is chosen. Other authors [4], [26], [9], recognizing that the problem does not have a unique solution, have tried to exploit information in the image to recover the *most likely* solution. Several authors [4], [9] have posed the problem in a probabilistic framework and, more recently, Sapiro [26], [25] has developed an algorithm based on the Probabilistic Hough Transform. The neural network approach [17] to color constancy can similarly be seen as a method of dealing with the inherent uncertainty in the problem. While these algorithms which model and work with uncertainty represent an improvement over earlier attempts at solving the color constancy problem, none of them can be considered the definitive solution. They are neither good enough to explain our own color constancy [3], nor are they good enough to support other visual tasks such as object recognition [16].

In Section 2, we address the limitations of existing algorithms by presenting a new illuminant estimation algorithm [20] within a general *correlation framework*. In this framework, illuminant estimation is posed as a correlation of the colors in an image and our prior knowledge about which colors can appear under which lights. The light that is most correlated with the image data is the most likely illuminant. Intuitively, this idea has merit. If the colors in an image "look" more yellow than they ought to, then one might assume that this yellowness correlated with a yellow illuminant. The correlation framework implements this idea in a three step process. First, in a preprocessing step, we code information about the interaction between image colors and illuminants. Second, we correlate this prior information with the information present in a particular image. That is, the colors in an image are used to derive a measure of the likelihood that each of the possible illuminants was the scene illuminant. Finally, these likelihoods are used to recover an estimate of the scene illuminant. We develop a particular instantiation of this framework—a new correlation algorithm—which has a number of attractive properties. It enforces the physical realisability of lights and surfaces, it is insensitive to spurious image colors, it is fast to compute, and it calculates the most likely answer. Significantly, we can also calculate the likelihood of all possible illuminants; effectively, we can return the best answer together with the error bars.

We further show (in Section 3) that the correlation framework we have developed is general and can be used to describe many existing algorithms. We will see how algorithms ranging from Gray-World [5] to Gamut Mapping [14], [10] to the Neural Network Approach [17], relate to different definitions of color, likelihood, and correlation; yet, in all cases, the same correlation calculation results. Moreover, by examining these algorithms in the same framework, we will come to understand how our new simple algorithm builds on and, more importantly, improves upon other algorithms. Furthermore, we will show that algorithms such as gamut mapping, previously criticized for their complexity, are, in fact, no more complex than the simplest type of color constancy computation.

Finally, in Section 4, we present experimental evidence to show that our new algorithm formulated in this framework does provide very good color constancy: It performs better than the other algorithms we tested.

2 Color by Correlation

We pose the color constancy problem as that of recovering an estimate of the scene illumination from an image of a scene taken under an unknown illuminant since, from this estimate, it is relatively straightforward to transform image colors to illuminant independent descriptors [19]. Here, we restrict attention to the case of an imaging system with three classes of sensor. In such a case, it is not possible to recover the full spectral power distribution of the illuminant; so, instead, an illuminant with spectral power distribution $E(\lambda)$ is characterized by $\underline{p}^E$: The response of the imaging device to an achromatic surface viewed under $E(\lambda)$. An estimate of the illuminant will accordingly be a 3-vector sensor response, $\hat{\underline{p}}^E$. However, as pointed out earlier, since we cannot recover the overall intensity of the illuminant, but only its chromaticity, we instead represent an illuminant by its 2D chromaticity vector $\underline{c}^E$, derived from the 3-vector of sensor responses by discarding intensity information (for example, using (3)). We represent surfaces in a similar fashion—specifically, we define surface color by the chromaticity of the response which the surface induces in a device when viewed under some canonical illuminant.

The chromaticity coordinates define a two-dimensional space of infinite extent. However, to help us formulate our solution to the illuminant estimation problem, we make two further assumptions. First, we assume that a given device will produce responses only within a finite region of this space—for a device such as a digital camera giving 8-bit data, this is clearly valid since sensor responses will be integers in the range 0 to 255 and, using (3), calculable chromaticity coordinates will be in the range $1/255$ to 1. Second, we assume that we can partition this space into $N \times N$ uniform regions. This assumption is justified on two grounds. First, all devices have some measurement error, which implies that all chromaticities within a region defined by this error must be considered equal. Second, for many applications, lights and surfaces with chromaticities within a certain distance of one another can effectively be considered to have the same color. Exactly how finely we need to partition this space—how big N should be—will depend on the application. We consider this issue later in the paper.

Partitioning the chromaticity space in this way implies that there are at most N^2 distinct illuminants and N^2 distinct surfaces—so that there can be at most N^2 distinct chromaticities in an image. In practice, the range of illuminants which we encounter in the world is much more restricted than this, so that the number (N_{ill}) of possible illuminants will be much smaller than N^2. For convenience, we define an $N_{ill} \times 2$ matrix C_{ill} whose $i\text{th}$ row is the chromaticity of the $i\text{th}$ illuminant. And, we use the notation C_{im} to represent the $N_{pix} \times 2$ matrix of image chromaticities.

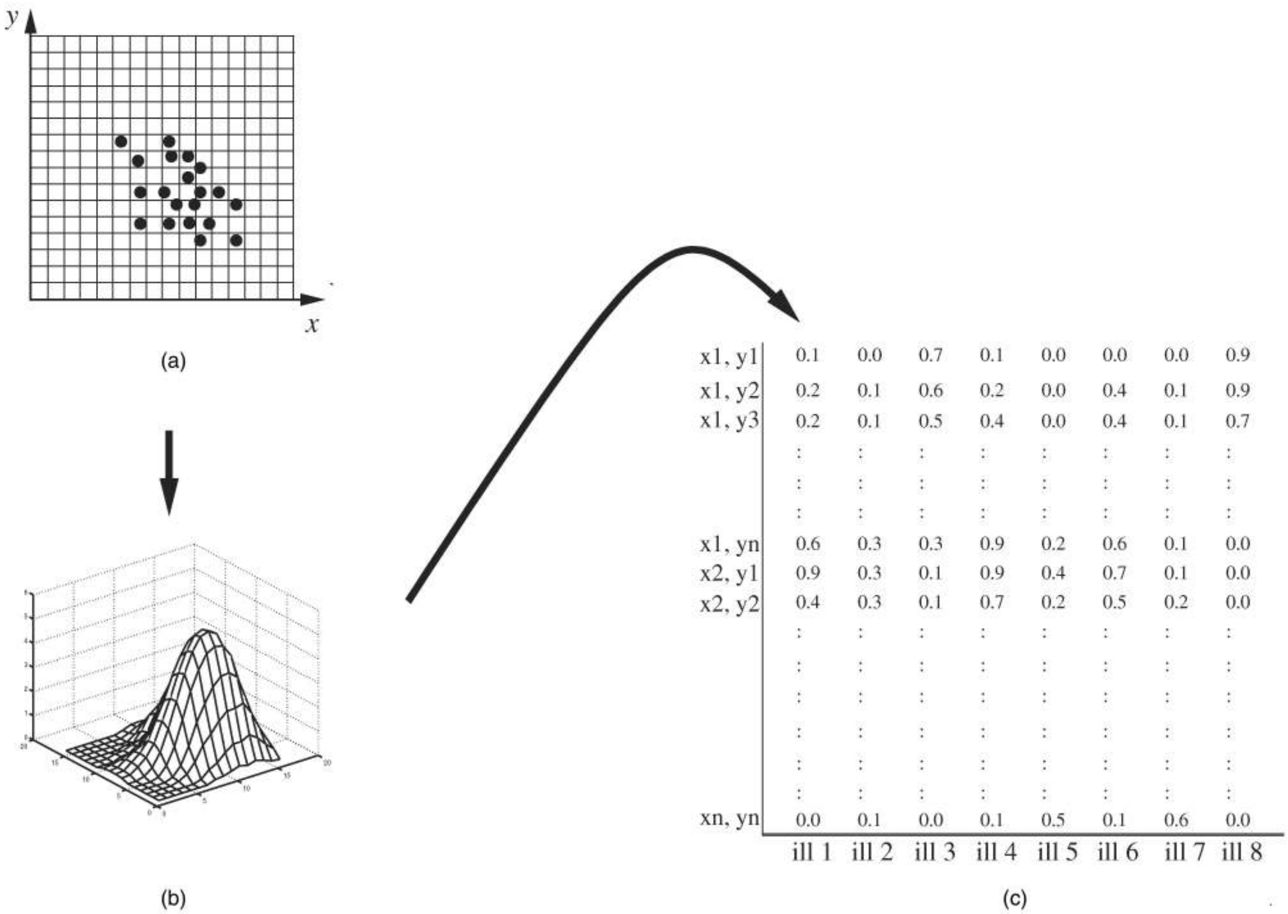

Fig. 1. Three steps in building a correlation matrix. (a) We first characterize which image colors (chromaticities) are possible under each of our reference illuminants. (b) We use this information to build a probability distribution for each light. (c) Finally, we encode these distributions in the columns of our matrix.

We now solve for color constancy in three stages. First, we build a correlation matrix to correlate possible image colors with each of the set of N_{ill} possible scene illuminants (see Fig. 1). For each illuminant, we characterize the range of possible image colors (chromaticities) that can be observed under that light (Fig. 1a). More details on how this can be done are given later in the paper. This information is used to build a probability distribution (Fig. 1b) which tells us the likelihood of observing an image color under a given light. The probability distributions for each light form the columns of a correlation matrix M (Fig. 1c) (each row of the matrix corresponds to one of the $N \times N$ discrete cells of the partitioned chromaticity space). Given a correlation matrix and an image whose illuminant we wish to estimate, we perform the following two steps (illustrated in Fig. 2). First, we determine which image colors are present in the image (Fig. 2a). This information is coded in a vector $\underline{v}$ of ones and zeros corresponding to whether or not a given chromaticity is present in the image. To this end, we define two operations $chist()$ and $thresh()$. The operation $chist()$ returns an $N^2 \times 1$ vector $\underline{h}$:

$$
\begin{aligned}
h(x_i * y_i) &= count_i / N_{pix} \\
count_i = \textstyle\sum_{j=1}^{N_{pix}} c_j, \qquad c_j &= \begin{cases} 1 & \text{if } C_{im}(j) = (x_i, y_i) \\ 0 & \text{otherwise.} \end{cases}
\end{aligned} \tag{4}
$$

that is, the ith element of $\underline{h}$ holds the number of times a chromaticity corresponding to (x_i, y_i) occurs in the image, normalized by N_{pix}, the total number of pixels in the image. For example, if chromaticity $(x_i, y_i)^t$ occurs 10 times in the image, then $\underline{h}(x_i * y_i) = 10/N_{pix}$. The second operation, $thresh(\underline{h})$, ensures that each image chromaticity is counted only once, regardless of the number of times it occurs in the image. Formally,

$$
thresh(x) = \begin{cases} 1, & \text{if } x > 0 \\ 0, & \text{otherwise.} \end{cases} \tag{5a}
$$

$$
\begin{aligned}
&thresh\big([h_1, h_2, \cdots, h_N]^t\big) = \\
&\qquad [thresh(h_1), thresh(h_2), \cdots, thresh(h_N)]^t.
\end{aligned} \tag{5b}
$$

With these definitions, $\underline{v}$ can be expressed:

$$
\underline{v} = thresh(chist(C_{im})). \tag{6}
$$

Then, we determine a measure of the correlation between this image data $\underline{v}$ and each of the possible illuminants. The usual expression of a correlation is as a vector dot-product. For example, if $\underline{a}$ and $\underline{b}$ are vectors, then they are strongly correlated if $\underline{a}.\underline{b}$ is large. We use a similar dot-product definition of correlation here. Each column of the correlation matrix M corresponds to a possible illuminant so that the elements of the vector returned by the product $\underline{v}^t M$ are a

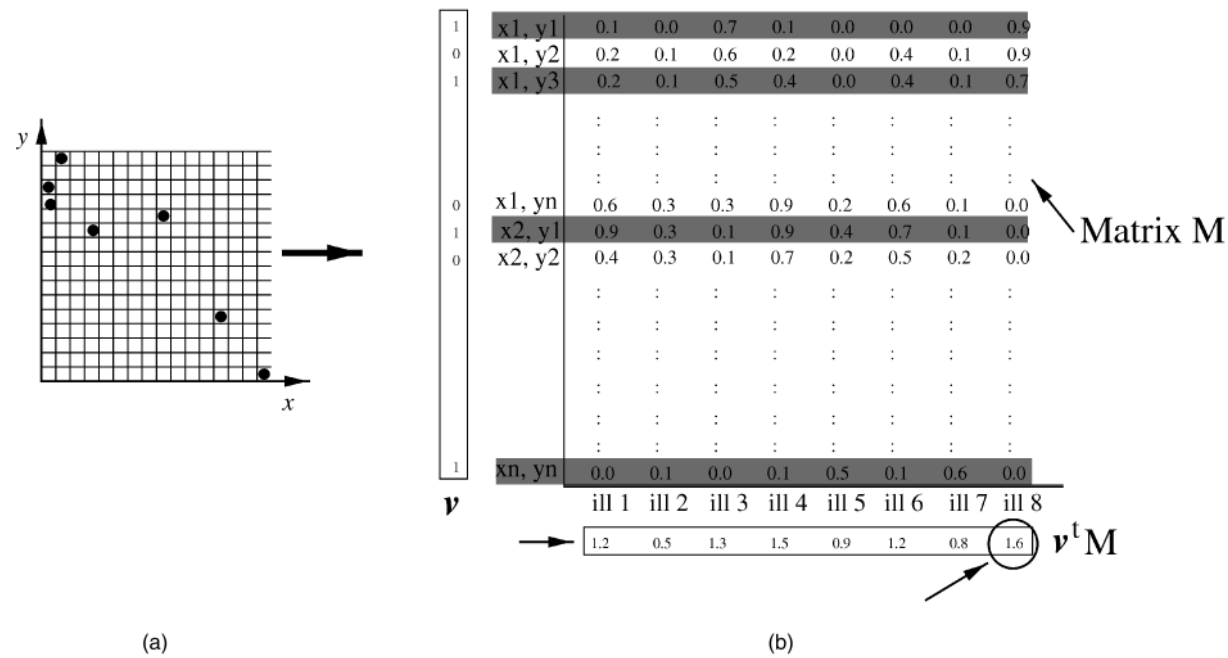

Fig. 2. Solving for color constancy in three stages. (a) Histogram the chromaticities in the image. (b) Correlate this image vector $\underline{v}$ with each column of the correlation matrix. (c) This information is used to find an estimate of the unknown illuminant, for example, the illuminant which is most correlated with the image data.

measure of how strongly the image data correlates with each of the possible illuminants. Fig. 2 is a graphical representation of this process. The highlighted rows of the correlation matrix correspond to chromaticities present in the image (entries of $\underline{v}$ which are one). To obtain a correlation measure for an illuminant, we simply sum the highlighted elements of the corresponding column. The result of this is a vector, $\underline{l}$ (Fig. 2b), whose elements express the degree of correlation of each illuminant—the bigger an element of this vector, the greater the correlation between the image data and the corresponding illuminant:

$$\underline{l} \;=\; \underline{v}^t M \;=\; thresh(chist(C_{im}))^t M. \qquad (7)$$

The final stage in solving for color constancy is to recover an estimate of the scene illuminant based on the correlation information (Fig. 2c). For example, we could choose the illuminant which is most highly correlated with the image data:

$$\hat{\underline{c}}^E = thresh2(thresh(chist(C_{im}))^t M)C_{ill}, \qquad (8)$$

where $thresh2()$ returns a vector with entries corresponding to:

$$thresh2(\underline{h}) \;=\; \underline{h}' \;\; h'_i = \begin{cases} 1, & \text{if } h_i = max(\underline{h}) \\ 0, & \text{otherwise.} \end{cases} \qquad (9)$$

Equation (8) represents our framework for solving color constancy. To completely specify the solution, however, we must define the entries of the correlation matrix M. Given a set of image data C_{im}, we would like to recover $Pr(E|C_{im})$—the probability that E was the scene illuminant given C_{im}. If we know the probability of observing a certain chromaticity $\underline{c}$, under illuminant E: $Pr(\underline{c}|E)$, then Bayes' rule [7] tells us how to calculate the corresponding probability $Pr(E|\underline{c})$: the probability that the illuminant was E, given that we observe chromaticity $\underline{c}$:

$$Pr(E|\underline{c}) = \frac{Pr(\underline{c}|E)Pr(E)}{Pr(\underline{c})}. \qquad (10)$$

Here, $Pr(E)$ is the probability that the scene illuminant is E, and $Pr(\underline{c})$ is the probability of observing the chromaticity $\underline{c}$, and the set of possible illuminants is defined by the $N_{ill} \times 2$ matrix C_{ill} and the range of possible chromaticities is defined by the $N \times N$ partitions of the chromaticity space. From (10), it follows that the probability that the illuminant was E, given the image data C_{im}, is given by:

$$Pr(E|C_{im}) = \frac{Pr(C_{im}|E)Pr(E)}{Pr(C_{im})}. \qquad (11)$$

Now, noting that for a given image $Pr(C_{im})$ is constant and if we assume that image chromaticities are independent, then we can rewrite (11) as:

$$Pr(E|C_{im}) = \left[\prod_{\forall \underline{c} \in C_{im}} Pr(\underline{c}|E)\right] Pr(E). \qquad (12)$$

Furthermore, if we assume that all illuminants are equally likely, then we have:

$$Pr(E|C_{im}) = k \prod_{\forall \underline{c} \in C_{im}} Pr(\underline{c}|E), \qquad (13)$$

where k is some constant. Of course, in general, it is not clear that all illuminants will occur with the same frequency; however, in the absence of information about the frequency of occurrence of individual lights, it makes sense to assume that all are equally likely. For some applications, it may be possible to provide such prior information and, in this case, with a slight modification to the framework, such information can be incorporated into the algorithm. For now, we simply assume that all lights are equally likely, but we also remind the reader that we are not considering all possible chromaticities as the set of potential illuminants, but, rather, we use a restricted set corresponding to "reasonable" lights.

Now, we define a likelihood function:

$$l(E|C_{im}) = \sum_{\forall \underline{c} \in C_{im}} \log(Pr(\underline{c}|E)) \tag{14}$$

and note that the illuminant which maximizes $l(E|C_{im})$ will also maximize $Pr(E|C_{im})$. The log-probabilities measure the correlation between a particular chromaticity and a particular illumination. We can then define a correlation matrix M_{Bayes} whose ijth entry is:

$$\log(Pr(\text{image chromaticity } \ i|\text{illuminant } \ j)).$$

It follows that the correlation vector $\underline{l}$ defined in (7) becomes the log-likelihood $l(E|C_{im})$:

$$l(E|C_{im}) = thresh(chist(C_{im}))^t M_{Bayes} \tag{15}$$

and our estimate of the scene illuminant can be written:

$$\hat{\underline{c}}^E = thresh2(thresh(chist(C_{im}))^t M_{Bayes})C_{ill}. \tag{16}$$

Equation (16) defines a well-founded maximum-likelihood solution to the illuminant estimation problem. It is important to note that, since we have computed likelihoods for all illuminants, we can augment the illuminant calculated in (16) with error bars. We could do this by returning the maximum-likelihood answer defined in (16) together with the likelihoods for each possible illuminant, defined in (15). These likelihoods tell us how much confidence we should have in the maximum-likelihood answer. Alternatively, rather than just returning the most-likely answer, we could return this answer together with other illuminants which had similar likelihoods. That is, we could return a set of plausible illuminants $C_{plausible}$ which we define:

$$C_{plausible} = diag(thresh3(thresh(chist(C_{im}))^t M_{Bayes}))C_{ill}, \tag{17}$$

where we replace the thresholding function $thresh2()$ used in (16) with a new function $thresh3()$, such that:

$$\begin{aligned} thresh3(x) &= 1, \quad \text{if } \ x \geq m, \\ thresh3(x) &= 0, \quad \text{otherwise} \end{aligned} \tag{18}$$

and m is chosen in an adaptive fashion such that $m \leq max(\underline{h})$. The function $diag(\underline{a})$ returns a diagonal matrix whose nonzero entries are the elements of $\underline{a}$ so that $C_{plausible}$ is an $N_{ill} \times 2$ matrix whose nonzero rows correspond to possible lights. This enables us to inform the user that the illuminant was, for example, either yellow Tungsten or cool white fluorescent, but that it was definitely not blue sky daylight. Once more, we can return $C_{plausible}$ together with the likelihoods of all the illuminants. We believe that this is a major strength inherent in our method and will be of significant value in other computer vision applications. For example, face tracking based on color can fail if the color of the skin (defined in terms of image colors) varies too much. In the context of the current paper, this is not a problem so long as the variation is explained by the color constancy computation.

3 Other Algorithms in the Framework

Equation (8) encapsulates our solution to the illuminant estimation problem. At the heart of this framework is a correlation matrix which encodes our knowledge about the interaction between lights and image colors. We show in this section that many existing algorithms can be reformulated in this same framework simply by changing the entries of the correlation matrix so as to reflect the assumptions about the interactions between lights and image colors made (implicitly) by these algorithms.

3.1 Gray-World

We begin with the so-called *Gray-World* algorithm which as well as being one of the oldest and simplest is still widely used. This algorithm has been proposed in a variety of forms by a number of different authors [5], [18], [21] and is based on the assumption that the spatial average of surface reflectances in a scene is achromatic. Since the light reflected from an achromatic surface is changed equally at all wavelengths, it follows that the spatial average of the light leaving the scene will be the color of the incident illumination. Buchsbaum [5] who was one of the first to explicitly make the gray-world assumption, used it, together with a description of lights and surfaces as low-dimensional linear models [6], to derive an algorithm to recover the spectral power distribution of the scene illuminant $E(\lambda)$ and the surface reflectance functions $S(\lambda)$. To recover an estimate of the scene illuminant in the form we require, that is, in terms of the sensor response of a device to the illuminant, is trivial, we simply need to take the average of all sensor responses in the image. That is:

$$\underline{p}^E = mean(RGB_{im}). \tag{19}$$

Equation (19) can equivalently be written as:

$$\underline{p}^E = hist(RGB_{im})^t \mathcal{I} RGB_{ill}, \tag{20}$$

where RGB_{ill} and RGB_{im}, respectively, characterize the set of all possible illuminants and the set of image pixels in camera RGB space (they are the 3D correlates of C_{ill} and C_{im} defined earlier). The operation $hist()$ is $chist()$ modified to work on RGBs rather than chromaticities and the matrix $\mathcal{I}$ is the identity matrix. In this formulation, $\mathcal{I}$ replaces the correlation matrix M_{Bayes} and, as before, can be interpreted as representing our knowledge about the interaction between image colors and surfaces. In this interpretation, the columns and rows of $\mathcal{I}$ represent possible illuminants and possible image colors, respectively. Hence, $\mathcal{I}$ tells us that given a sensor response $\underline{p}$ in an image, the only illuminant consistent with it is the illuminant characterized by the same sensor response. Correspondingly, the vector

$$\underline{l} = hist(RGB_{im})^t \mathcal{I} \tag{21}$$

whose elements contain the number of image colors consistent with each illuminant, can be interpreted as a measure of the likelihood that each illuminant (each distinct RGB present in the image) is the scene illuminant. Based on these likelihoods, we calculate an estimate of the scene illuminant by taking the weighted average of all illuminants. For example, if $(200, 10, 200)^t$ appears in an image, then $(200, 10, 200)^t$ could be the illuminant color. Equally, if $(100, 50, 20)^t$ appears in the image, it too is a candidate illuminant. If both image colors appear in the image, then according to $\mathcal{I}$ they are both possible candidates for the illuminant. The mean estimate of the two provides a well-founded statistical estimate of the illuminant relative to the priors we have used (the diagonal matrix $\mathcal{I}$). In fact, this example highlights one of the problems with using these priors since, according to our model of image formation ((1)), if either of these RGBs is considered the RGB of the illuminant, then the other cannot possibly be so.

While it is often used for color constancy, the Gray-World algorithm has a number of limitations. First, Gershon et al. [18] have pointed out that the spatial average computed in (19) is biased towards surfaces of large spatial extent. They proposed a modified algorithm which alleviates this problem by segmenting the image into patches of uniform color prior to estimating the illuminant. The sensor response from each segmented surface is then counted only once in the spatial average, so that surfaces of different size are given equal weight in the illuminant estimation stage. It is trivial to add this feature in our framework; we simply need to apply the thresholding operation $thresh()$, defined in (5), to the output from $hist()$:

$$\underline{p}^E = thresh(hist(RGB_{im})^t)\mathcal{I}RGB_{ill}. \tag{22}$$

A second limitation of the Gray-World algorithm is highlighted by (20)—the identity matrix $\mathcal{I}$ does not accurately represent our knowledge about the interaction of lights and surfaces. Improving color constancy then, amounts to finding matrices such as M_{Bayes} defined above which more accurately encode that knowledge.

3.2 3D Gamut Mapping

The matrix $\mathcal{I}$ tells us that a reddish RGB observed in an image is only consistent with an illuminant of that color. In fact, such an RGB is consistent with both a red surface under a white light and a white surface under a red light, and with many other combinations of surface and illuminant. Forsyth [14] developed an algorithm, called CRULE to exploit this fact. CRULE is founded on the idea of color gamuts: The set of all possible sensor responses observable under different illuminants. Forsyth showed that color gamuts are closed, convex, bounded, and that, most importantly, each is a strict subset of the set of possible image colors. The gamut of possible image colors for a light can be determined by imaging all possible surfaces (or a representative subset thereof) under that light. We can similarly determine gamuts for each of our possible illuminants, i.e., for each row of RGB_{ill}, and can code them in a correlation matrix. That is, we define a matrix M_{For}, such that if image color i can be observed under illuminant j, then we put a one in the ijth entry of M_{For}, otherwise, we put a zero. The matrix M_{For} more accurately represents our knowledge about the world and can be used to replace $\mathcal{I}$ in (20).

From the colors present in an image, CRULE determined a set of feasible illuminants. An illuminant is feasible if all image colors fall within the gamut defined by that illuminant. In our framework, the number of image colors consistent with each illuminant can be calculated:

$$\underline{l} = thresh(hist(RGB_{im})^t)M_{For}, \tag{23}$$

where $thresh()$ and $hist$, as defined previously and used together, ensure that each distinct image color is counted only once. If there are N_{surf} distinct surfaces in an image, then any illuminant corresponding to entries of $\underline{l}$ which are equal to N_{surf} are consistent with all image colors and are therefore feasible illuminants.

Once the set of feasible illuminants has been determined, the final step in CRULE is to select a single illuminant from this set as an estimate of the unknown illuminant. Previous work has shown [2] that the best way to do this is to take the mean of the feasible illuminants. In our framework, this can be achieved by:

$$\hat{\underline{p}}^E = thresh2(thresh(hist(RGB_{im}))^t M_{For})RGB_{ill}, \tag{24}$$

where $thresh2()$ is defined as before. Equation (24), though equivalent to CRULE (with mean selection), is a much simpler implementation of it. In Forsyth's CRULE, the notion of which image colors can appear under which lights was modeled analytically as closed continuous convex regions of RGB space. Also, the illuminants themselves were not represented explicitly. Rather, an illuminant $\underline{p}^o$ is defined by the mapping (actually a 3×3 diagonal matrix) that takes responses observed under $\underline{p}^o$ to reference or canonical lighting conditions $\underline{p}^c$. In CRULE, computation involves calculating mapping sets for each image color and then intersecting these sets to arrive at the overall plausible set (which contains those mappings that take all image colors to canonical counterparts).

Computation aside, our new formulation has another significant advantage over CRULE. Since rather than saying that an illuminant is possible if and only if it is consistent with all image colors, we can instead look for illuminants that are consistent with most image colors. This subtle change cannot be incorporated easily into the CRULE algorithm, yet it is important that it is since, in CRULE, if no illuminant is globally consistent, there is no solution to color constancy. To implement majority consistency in our framework is straightforward. We simply replace $thresh2()$ with the thresholding function $thresh3()$ defined in (18). Thus, the improved CRULE algorithm, can be written as:

$$\hat{\underline{p}}^E = thresh3(thresh(hist(RGB_{im}))^t M_{For})RGB_{ill}. \tag{25}$$

3.3 Color In Perspective (2D Gamut Mapping)

While the formulation of Forsyth's CRULE algorithm given above addresses some of its limitations there are other problems with CRULE which this formulation doesn't resolve. First, Finlayson [10] recognized that features such as shape and shading affect the magnitude of the recovered light, but with any significance, not its color. To avoid calculating the intensity of the illuminant (which cannot be recovered [23]), Finlayson carried out computation in a 2D chromaticity space. Once again, if we characterize image colors and illuminants by their chromaticities, we can define a new matrix, M_{Fin} whose ijth element will be set to one when chromaticity i can be seen under illuminant j and to zero, otherwise. We can then substitute M_{Fin} in (8):

$$\underline{\hat{c}}_E = thresh2(thresh(chist(C_{im}))^t M_{Fin})C_{ill}. \tag{26}$$

Assuming the thresholding operations $thresh()$ and $thresh2()$ are chosen as for Forsyth's algorithm (we could, of course, use $thresh3()$ instead of $thresh2()$ if we wanted to implement majority consistency), then the illuminant estimate $\underline{\hat{c}}^E$ is the averaged chromaticity of all illuminants consistent with all image colors. Previous work [12] has shown however, that the mean chromaticity is not the best estimate of the illuminant and that the chromaticity transform should be reversed before the averaging operation is performed. This can be achieved here by defining a matrix RGB^N_{ill}, whose ith row is the ith row of RGB_{ill} normalized to unit length. The illuminant estimate is now calculated:

$$\underline{p}^E = thresh2(thresh(chist(C_{im}))^t M_{Fin})RGB^N_{ill}. \tag{27}$$

Another problem with gamut mapping is that not all chromaticities correspond to plausible illuminants (for example, purple lights do not occur in practice). This observation is also simple to implement since we can simply restrict the columns of M_{Fin} to those corresponding to plausible lights.

3.4 Illuminant Color by Voting

Sapiro [26], [25] has recently proposed an algorithm for estimating the scene illuminant which is based on the Probabilistic Hough Transform. In this work, Sapiro represents lights and surfaces as low-dimensional linear models and defines, according to this model, a probability distribution from which surfaces are drawn. Given a sensor response from an image, a surface is selected according to the defined distribution. This surface, together with the sensor response, is used to recover an illuminant. If the recovered illuminant is a feasible illuminant (in Sapiro's case an illuminant on the daylight locus), a vote is cast for that illuminant. For each sensor response, many surfaces are selected and so many votes are cast. To get an estimate of the illuminant, the cumulative votes for each illuminant are calculated by summing the votes from all sensor responses in the image. The illuminant with maximum votes is selected as the scene illuminant.

The votes for all illuminants for a single sensor response $\underline{p}$ represent an approximation to the probability distribution: $Pr(E|\underline{p})$—the conditional probability of the illuminant given the observed sensor response. Sapiro chooses the illuminant which maximizes the function:

$$\sum_{p \in RGB_{im}} Pr(E|\underline{p}). \tag{28}$$

Since we know the range of possible image colors, rather than compute the probability distributions $Pr(E|\underline{p})$ on a per image basis, we could instead, using Bayes rule, compute them once for all combinations of sensor responses and illuminants. We can then define a matrix M_{Sapiro} whose ijth entry is $Pr(\text{illuminant } j|\text{image color } i)$, which, by Bayes rule, is proportional to the probability of observing image color i under illuminant j. It then follows that Sapiro's estimate of the illuminant can be found in our framework by:

$$\underline{\hat{p}}^E = thresh3(hist(RGB_{im})^t M_{Sapiro})RGB_{ill}, \tag{29}$$

where the matrix M_{Sapiro} is equal to $ke^{M_{Bayes}}$. We note that in (29) the image histogram is not thresholded so that Sapiro's algorithm, as Gray-World, will be sensitive to large areas of uniform color.

3.5 Probabilistic Algorithms

Brainard and Freeman [4] have recently given a Bayesian formulation of the color constancy problem. Their approach is again founded on a linear models representation of lights and surfaces. That is, each light and surface is represented by a weighted sum of a small number of basis functions so that these weights are sufficient to define a light or surface. Principal component analyses of collections of surfaces and illuminants were used to determine suitable basis functions and the corresponding weights for each light and surface. The authors then defined probability distributions for these weights and used Bayesian decision theory to recover estimates of the weights for the surfaces and illuminant in an image. So, if $\underline{x}$ represents the combined vector of weights for all the surfaces and the light in an image, the problem is to estimate $\underline{x}$.

If there are N_{surf} surfaces in the image, then the vector to be recovered is $(3N_{surf} + 3)$-dimensional. Estimating $\underline{x}$ is therefore computationally extremely complex. The authors have implemented the algorithm as a numerical search problem and shown results for the case $N_{surf} = 8$. However, since typical images contain many more surfaces than eight, as a practical solution for color constancy, this approach is far too complex. A precise formulation of their algorithm is not possible within our framework; however, we can use their prior distributions on surfaces and illuminants when constructing our correlation matrix. If we then restrict the problem to that of recovering an estimate of the unknown illuminant, then our approach should produce similar results.

An approach which is much closer to the algorithm we have presented was proposed by D'Zmura and Iverson [9]. They also adopted a linear models representation of surfaces, but they used these models to derive a likelihood distribution $Pr((x,y)|E(\lambda))$. That is the probability of observing a given CIE-xy chromaticity [31] coordinate, under an illuminant $E(\lambda)$. This is done by first defining distribution functions for the weights of their linear model of surfaces. Then, they generated a large number of surfaces by selecting weights according to these distributions and calculated the corresponding chromaticity coordinates for these surfaces. By selecting a large number of surfaces, a good approximation to $Pr((x,y)|E(\lambda))$ can be found. If we put likelihoods corresponding to these probabilities in a correlation matrix, then this algorithm can be formulated in the framework we have developed. We point out that this algorithm, like Gray-World, takes no account of the relative frequency of individual chromaticities: The function

$thresh()$ is not used. As such, the algorithm is again highly sensitive to large image areas of uniform color and, so, can suffer serious failures.

3.6 Neural Networks

The final algorithm we consider is the Neural Network approach of Funt et al. [17]. Computation proceeds in three stages and is summarized below:

$$\begin{aligned} output_1 &= thresh4(thresh(hist(C_{im})^t)M_{Funt}) \\ output_2 &= thresh4(output_1^t M_{Funt,1}) \\ output_3 &= output_2^t M_{Funt,2}, \end{aligned} \tag{30}$$

where $thresh4$ is similar to $thresh3$ but its exact definition has not been specified [17]. In the parlance of Neural Networks, $output_1$ is the first stage in a three layer perceptron calculation. The second, hidden layer computation, is modeled by the second equation and the final output of the Neural Net is $output_3$. The correlation matrix, $M_{Funt,1}$, typically has many fewer columns than M_{Funt}. Moreover, $M_{Funt,2}$ only has two columns and, so, the whole network only outputs two numbers. These two numbers are trained to be the chromaticity of the actual scene illuminant. As such, we can replace $M_{Funt,2}$ by C_{ill} (though it is important to realize that here C_{ill} is discovered as a result of training and does not bear a one to one correspondence with actual illuminants).

In the context of this paper, $output_1$ is very similar to (8) albeit with a different correlation matrix and slightly different threshold functions. The other two stages address the question of how a range of possible illuminants is translated into a single illuminant chromaticity estimate. As one might imagine, the Neural Net approach, which basically fits a parametric equation to model image data, has been shown to deliver reasonably good estimates. However, unlike the approach advocated here, it is not possible to give certainty measures with the estimate nor is it possible to really understand the nature of the computation that is taking place.

4 Results

We conducted two experiments to assess the performance of our new correlation algorithm and to compare it to existing algorithms. First, to get some idea of the algorithm's performance over a large data set, we tested it on synthetically generated images. In a second experiment, we tested the algorithm on a number of real images captured with a digital camera. We show exemplar results of these tests for a small number of images.

In the experiments on synthetic images, we tested five algorithms: Gray-World, Modified Gray-World, 2D Gamut Mapping (with mean selection), Sapiro's algorithm, and the new Color by Correlation algorithm described in this paper. Gray-World simply uses the average of all the sensor responses in the image as an estimate of the unknown illuminant. Modified Gray-World is similar, except that each distinct sensor response is counted only once when forming the average, regardless of how many times it occurs in the image. More sophisticated segmentation algorithms could be used, as Gershon et al. [18] suggest; however, the question of how best to segment is still open and, so, we adopt the simplest approach here. Two-dimensional Gamut Mapping is Finlayson's Color In the Perspective algorithm, formulated in the correlation framework and Sapiro's algorithm is also reformulated in this same framework.

Before running the correlation-based algorithms, we must make the correlation matrices themselves. In the case of the 2D Gamut Mapping (matrix M_{Fin}), we want to determine the range of image chromaticities that are possible under each of the illuminants between which we wish to distinguish. One way to obtain this information would be to take the camera and capture a wide range of surface reflectances under each of the lights between which we wish to distinguish. In this way, we can define the gamut of colors which the camera records under each light. However, this approach is somewhat cumbersome especially if the number of possible lights is large. Fortunately, given some knowledge about our camera, we can instead generate these gamuts using synthetic data. Specifically, if we know the spectral response characteristics of the camera, then we need only measure the surface reflectances of a range of objects and the spectral power distribution of each illuminant between which we wish to distinguish. We can then use (2) to generate the set of possible sensor responses under each illuminant and, from these, we can calculate the corresponding chromaticities. We then take the convex hull of these chromaticities and consider that any chromaticity within the hull is part of the gamut (it's corresponding entry in the correlation matrix is one) and that all other chromaticities are outside it (their corresponding entries are zero [14]).

We point out that, in using (2) to generate the sensor responses, we are assuming that the camera has a linear response, whereas, in practice, this is often not the case. In such cases, we must account for any nonlinearity in the camera's response when calculating the responses—that is, we must characterize the nature of the camera's nonlinearity and modify (2) to take account of it. Alternatively, we could calculate the matrices using the assumption of linear data and then linearize the data recorded by the camera before applying the illuminant estimation algorithm.

The entries of the matrix M_{Bayes} can also be found using synthetic responses generated using (2). However, now, rather than recording only whether a chromaticity is possible or not, we want to record the relative frequency with which it occurs. To do this, we can use the same set of surface reflectances to calculate chromaticity coordinates as before, then, to estimate the probability of a given image color, we simply count the number of surfaces falling in each bin of the discretized chromaticity space. The entries of M_{Bayes} are the $\log$ of these raw counts normalized by the total number of chromaticities. The matrix M_{Sapiro} is created in the same way except that we put actual probabilities rather than log probabilities in the matrix.

When calculating the correlation matrices, we must decide how to partition the chromaticity space. This partitioning will depend both on the characteristics of the camera's sensors and also on which chromaticity space is being used. There are many chromaticity spaces which could potentially be employed; for the experiments reported here we used the coordinates:

$$c_1 = \left(\frac{p_1}{p_2}\right)^{\frac{1}{3}}, \quad c_2 = \left(\frac{p_3}{p_2}\right)^{\frac{1}{3}} \tag{31}$$

since this leads to chromaticities which are reasonably uniformly distributed. This has the advantage that a simple uniform discretization of the space can be employed. The method is quite insensitive to the level of discretization of the space—we have achieved good results using discretizations ranging from 16×16 to 64×64. The results reported here are based on a 24×24 discretization of the space which we have found to work well for a range of cameras.

We tested the algorithms' performance using images synthesized using (2). For sensor response curves, we used monochromator measurements of a HP Photosmart C500 digital camera. These curves were then used to generate the correlation matrices. Key to the success of the method is the choice of lights and surfaces used to build the correlation matrices. For the experiments reported here, we wanted to generate a matrix which would give good results for a wide range of illuminants, so we chose a set of 37 lights, representing a range of commonly occurring indoor and outdoor illumination. The set includes daylights with correlated color temperatures ranging from D75 to D40, Planckian blackbody radiators, ranging from 3,500K to 2,400K, and a variety of fluorescent sources. The set of surface reflectances we use is most important when we test the algorithm on real images since how well the distribution of these surfaces matches the distribution of reflectances in the real world will have a large effect on the success of the algorithm. For the synthetic experiments, it is enough to choose a wide range of reflectance functions. We used two sets: a set of 462 Munsell chips [31] and a collection of object surface reflectances measured by Vrhel et al. [30].

To create the synthetic images in which we tested the algorithms, we randomly selected between 2 and 64 surfaces from a set of surface reflectances and a single illuminant, drawn from the set of 37. To make the test more realistic we used reflectances from a set of natural surface reflectances measured by Parkkinen et al. [24] rather than using the same reflectances on which the correlation matrices were built. We calculated the sensor response for a surface and then weighted each surface by a random factor chosen to ensure that the number of pixels in each image was 512×512. Since many real images consist of large areas of uniform color (for example, outdoor scenes often contain large regions of blue sky), to make the images more realistic, the factors were chosen such that one surface always occupied at least 40 percent of the image.

To determine an algorithm's estimate of the illuminant, we simply calculate an image histogram, and then the likelihood for each illuminant according to (7). These likelihoods are then used to select a single illuminant from the set as an estimate of the scene illuminant. For the 2D Gamut Mapping algorithm, we used the mean selection method [12]; whereas, for Sapiro's algorithm and the new Color by Correlation approach, we chose the illuminant with maximum likelihood.

To assess the relative performance of the algorithms, we chose a root mean square error (RMSE) measure—specifically, the chromaticity error between the image under D65 illumination and an estimate (calculated using each algorithm's estimate of the illumination) of the image under D65. RMSE is commonly used in the computational color constancy literature [2], [17] and, while it is not the most intuitive error measure (it is not easy to interpret what a given RMS error means in terms of visual difference between images), it at least allows the relative performance of the algorithms to be assessed. To help give some idea of what a certain RMS error corresponds to in terms of visual difference, in the second experiment, on real images, we give RMS errors for a number of corrected images, together with a print of those images. RMS error is calculated as follows: Each of the algorithms was used to generate an estimate of the unknown scene illuminant in terms of a 2D chromaticity coordinate. We used this estimate to rerender the image as it would have appeared under standard daylight D65 illumination. Rerendering was performed by finding the diagonal mapping which takes the algorithm's estimate of the scene illuminant to the chromaticity of D65 illumination. We then applied this mapping to all RGBs in the image to obtain an estimate of the scene as it would have appeared under D65 illumination. We also performed a similar correction using a mapping from the chromaticity of the actual scene illuminant to D65 illumination. We then calculated the root mean square error in chromaticity space between these two images. Since algorithm performance is measured in a 2D chromaticity space and since the Gray-World algorithms work in 3D sensor space, it might be thought that the algorithms which set out to recover a 2D estimate would have an unfair advantage over the 3D algorithms. We tested this theory by modifying the Gray-World algorithms to work in 2D chromaticity space and found that this, in fact, led to worse performance than in the 3D case. For this reason, only the 3D results are reported here.

Fig. 3 shows the relative performance of the five algorithms in terms of the average RMS chromaticity error against the number of surfaces in the image. Results were calculated for images with 2, 4, 8, 16, 32, and 64 surfaces and, in each case, the average was taken over 500 images. We can draw a number of conclusions from these results. First, accurately encoding information about the world leads to improved color constancy performance; the gamut-mapping algorithm and the two algorithms exploiting probability information all perform considerably better than the gray-world algorithms. Further, we can see that adding information about the probabilities of image colors under different illuminants further improves performance. It is important though, that this information is encoded correctly. Our new algorithm, which correctly employs Bayes's rule, gives a lower average RMSE than the second best algorithm. However, Sapiro's algorithm which does not correctly encode probability information performs slightly worse than the 2D gamut-mapping algorithm.

Previous work [12] has demonstrated that 2D gamut mapping produces better results than most other algorithms [13], [12]. So, it is significant that our new approach delivers much better constancy. Moreover, the Neural Net approach (for which there insufficient information for us to implement) has also been shown [17] to perform similarly to 2D gamut mapping. Thus, on the basis of the results

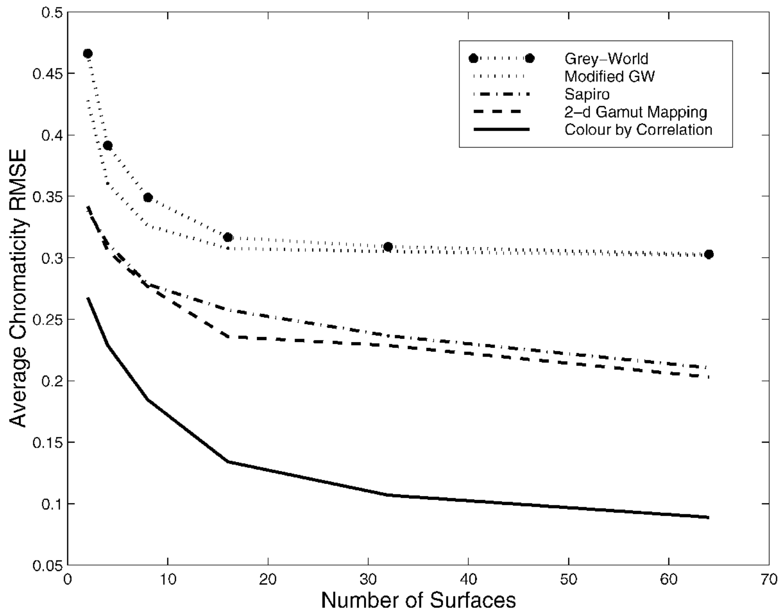

Fig. 3. Average RMSE chromaticity error for: gray world (dotted line with *), Modified Gray World (dotted line), Sapiro's algorithm (dash-dot line), 2D gamut mapping (dashed line), and Color by Correlation (solid line).

presented here, it is fair to say that, to our knowledge, the new algorithm outperforms all other algorithms. The second experiment we ran was to test the performance of the algorithm on real images. We used images from two different digital cameras: a HP-Photosmart C500 and a prototype digital still camera based on a Sony ICX085 CCD. Both cameras were modified so that they gave raw sensor data and were calibrated to ensure that this data was linearly related to incident light. Thus, (2) is an accurate model of image formation. We measured the spectral sensitivity curves of both devices using a monochromator and used these measurements when building the correlation matrices for the various algorithms.

The raw data from the camera was averaged down by a factor of five (in width and height) and this averaged image was used as input to the illuminant estimation algorithms. Using each algorithm's estimate of the scene illuminant, we rerendered the full-size captured image to D65 illumination, following the procedure described above. Figs. 4 and 5, and Table 1 show typical examples of the algorithm's

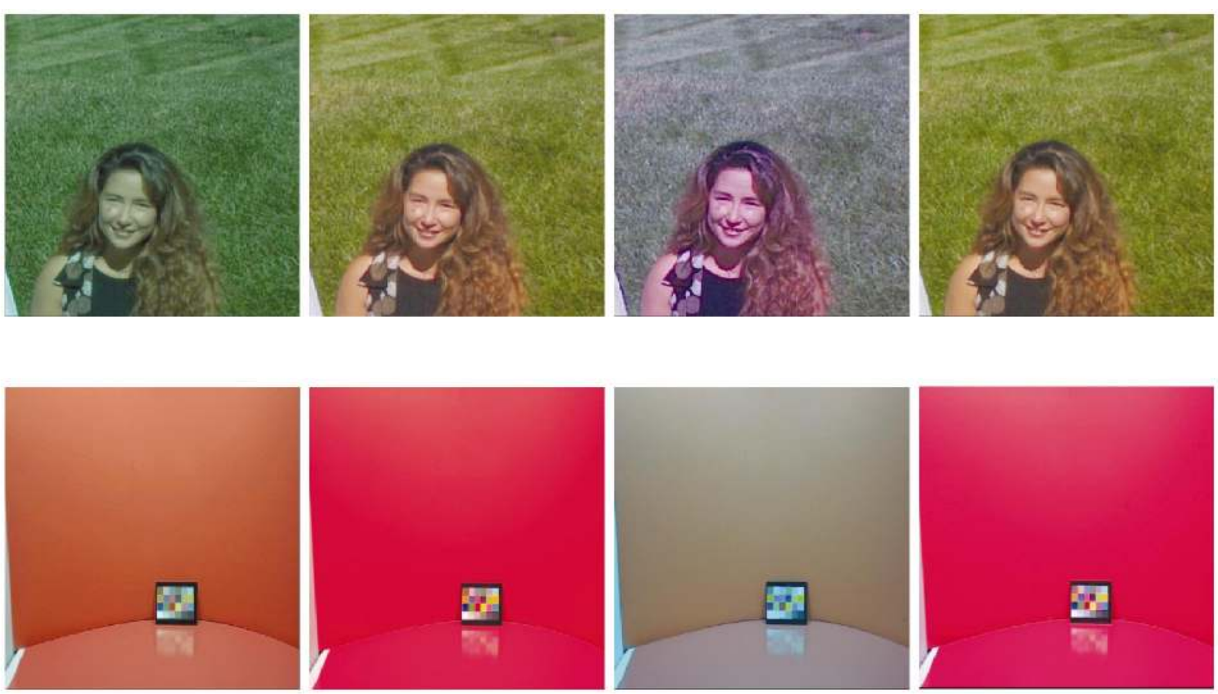

Fig. 4. Left to right: raw camera image, correction based on; measured illuminant, Gray-World, and Color by Correlation. Images were taken under daylight D50 (top) and simulated D65 (bottom).

Fig. 5. Left to right: raw camera image, correction based on; measured illuminant, 2D Gamut Mapping, and Color by Correlation. Images were taken under illuminant A (top) and cool white fluorescent (bottom).

performance for the two cameras. Fig. 4 shows results for two scenes captured with the first camera. For each scene, we show four images; the raw data as captured by the camera, the image rerendered to D65 illumination using a spectral measurement of the scene illuminant (obtained by placing a white tile in the scene and measuring the spectrum of the reflected light), the image rerendered using the illuminant estimate recovered by our new algorithm, and the image rerendered using the Gray-World estimate. It is clear that the image rerendered using the new algorithm's estimate of the illuminant is a very close match to that obtained using the measured illuminant. In contrast, the performance of Gray-World is worse and while, in terms of RMSE (see Table 1), as it is better than doing no correction the images are visually a very poor match to the properly corrected image. Fig. 5 shows results for the second camera. Here, rather than comparing performance with the Gray-World algorithm, we show how well the 2D Gamut Mapping algorithm performs (the second best algorithm in the experiments with synthetic images). The 2D Gamut Mapping algorithm does perform better than the Gray-World approach; however, performance is still some way behind that which can be obtained using the new algorithm. Table 1 summarizes the algorithm performance for the four images in terms of the RMS error measurement used in the synthetic experiments. Again, using this measurement, the new algorithm performs very well and better than the other algorithms tested. Four images do not represent an exhaustive test of the algorithm's performance and the results presented here are intended only to give an idea of the kind of the performance that can be obtained with the various algorithms. Though the results are typical of the performance we have achieved with the new algorithm, we are currently in the process of compiling a database of images on which to more thoroughly test performance.

TABLE 1
Average Root Mean Square Error between Images Corrected to the D65 Illumination Using an Estimate of the Scene Light and Images Corrected Using a Measurement of the Scene Light

	No CC	G-W	C by C	2-d G-M
Average	0.580	0.49	0.11	0.21

5 Conclusions

In this paper, we have considered the color constancy problem; that is how we can find an estimate of the unknown illuminant in a captured scene. We have seen that existing constancy algorithms are inadequate for a variety of reasons. For example, many of them make unrealistic assumptions about images or their computational complexity is such that they are unsuitable as practical solutions to the problem. Here, we have presented a correlation framework in which to solve for color constancy. The simplicity, flexibility, and robustness of this framework makes solving for color constancy easy (in a complexity sense). Moreover, we have shown how a particular Bayesian instantiation of the framework leads to excellent color constancy (better than other algorithms tested). A number of other previously proposed algorithms were also placed within the correlation framework, and others which, while they cannot be precisely formulated within the framework, were shown to be closely related to it.

Acknowledgments

This work was supported by Hewlett-Packard Incorporated and the EPSRC.

References

[1] L. Arend and A. Reeves, "Simultaneous Color Constancy," *J. Optical Soc. Am., A,* vol. 3, no. 10, pp. 1743-1751, 1986.

[2] K. Barnard, "Computational Color Constancy: Taking Theory into Practice," master's thesis, Simon Fraser Univ., School of Computing Science, 1995.

[3] D.H. Brainard, W.A. Brunt, and J.M. Speigle, "Color Constancy in the Nearly Natural Image. I. Asymmetric Matches," *J. Optical Soc. Am., A,* vol. 14, no. 9, pp. 2091-2110, 1997.

[4] D.H. Brainard and W.T. Freeman, "Bayesian Color Constancy," *J. Optical Soc. Am., A,* vol. 14, no. 7, pp. 1393-1411, 1997.

[5] G. Buchsbaum, "A Spatial Processor Model for Object Colour Perception," *J. Franklin Inst.,* vol. 310, pp. 1-26, 1980.

[6] J. Cohen, "Dependency of the Spectral Reflectance Curves of the Munsell Color Chips," *Psychonomic Science,* vol. 1, pp. 369-370, 1964.

[7] R.O. Duda and P.E. Hart, *Pattern Classification and Scene Analysis.* John Wiley and Sons, 1973.

[8] M.M. D'Zmura and G. Iverson, "Color Constancy. I. Basic Theory of Two-Stage Linear Recovery of Spectral Descriptions for Lights and Surfaces," *J. Optical Soc. Am., A,* vol. 10, no. 10, pp. 2148-2165, 1987.

[9] M.M. D'Zmura and G. Iverson, "Probabilistic Color Constancy," *Geometric Representations of Perceptual Phenomena: Papers in Honor of Tarow Indow's 70th Birthday,* M.M. D'Zmura, D. Hoffman, G. Iverson, and K. Romney, eds., Laurence Erlbaum Assoc., 1994.

[10] G.D. Finlayson, "Color in Perspective," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 18, no. 10, pp. 1034-1038, Oct. 1996.

[11] G.D. Finlayson and B.V. Funt, "Color Constancy with Shadows," *Perception,* vol. 23, pp. 89-90, 1994.

[12] G. Finlayson and S. Hordley, "A Theory of Selection for Gamut Mapping Colour Constancy," *Proc. Computer Vision and Pattern Recognition '98,* pp. 60-65, June 1998.

[13] G.D. Finlayson, P.M. Hubel, and S. Hordley, "Color by Correlation," *Proc. Fifth Colour Imaging Conf.,* pp. 6-11, Nov. 1997.

[14] D.A. Forsyth, "A Novel Algorithm for Colour Constancy," *Int'l J. Computer Vision,* vol. 5, no. 1, pp. 5-36, 1990.

[15] B.V. Funt, M.S. Drew, and J. Ho, "Color Constancy from Mutual Reflection," *Int'l J. Computer Vision,* vol. 6, pp. 5-24, 1991.

[16] B. Funt, K. Barnard, and L. Martin, "Is Machine Colour Constancy Good Enough?" *Proc. Fifth European Conf. Computer Vision,* pp. 455-459, June 1998.

[17] B.V. Funt, V. Cardei, and K. Barnard, "Learning Color Constancy," *Proc. Fourth Color Imaging Conf.,* pp. 58-60, Nov. 1996.

[18] R. Gershon, A.D. Jepson, and J.K. Tsotsos, "From [R,G,B] to Surface Reflectance: Computing Color Constant Descriptors in Images," *Perception,* pp. 755-758, 1988.

[19] P.M. Hubel, J. Holm, G.D. Finlayson, and M.S. Drew, "Matrix Calculations for Digital Photography," *Proc. Fifth Color Imaging Conf.,* pp. 105-111, Nov. 1997.

[20] P.M. Hubel and G.D. Finlayson, "Whitepoint Determination Using Correlation Matrix Memory," US Patent Application, Submitted.

[21] E.H. Land, "The Retinex Theory of Color Constancy," *Scientific Am.,* pp. 108-129, 1977.

[22] H.-C. Lee, E.J. Breneman, and C.P. Schulte, "Modeling Light Reflection for Computer Color Vision," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 12, no. 4, pp. 402-409, 1986.

[23] L.T. Maloney and B.A. Wandell, "Color Constancy: A Method for Recovering Surface Spectral Reflectance," *J. Optical Soc. Am., A,* vol. 3, no. 1, pp. 29-33, 1986.

[24] J. Parkkinen, T. Jaaskelainen, and M. Kuittinen, "Spectral Representation of Color Images," *Proc. IEEE Ninth Int'l Conf. Pattern Recognition,* vol. 2, pp. 933-935, Nov. 1998.

[25] G. Sapiro, "Color and Illuminant Voting," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 10, pp. 210-218, 1985.

[26] G. Sapiro, "Bilinear Voting," *Proc. Int'l Conf. Computer Vision '98,* pp. 178-183, Nov. 1998.

[27] S.A. Shafer, "Using Color to Separate Reflection Components," *Color Research and Application,* vol. 21, no. 11, pp. 1210-1215, 1999.

[28] M.J. Swain and D.H. Ballard, "Color Indexing," *Int'l J. Computer Vision,* vol. 7, no. 1, pp. 11-32, 1991.

[29] S. Tominaga, "Surface Reflectance Estimation by the Dichromatic Model," *Color Research and Application,* vol. 21, no. 2, pp. 104-114, 1996.

[30] M.J. Vrhel, R. Gershon, and L.S. Iwan, "Measurement and Analysis of Object Reflectance Spectra," *Color Research and Application,* vol. 19, no. 1, pp. 4-9, 1994.

[31] G. Wyszecki and W.S. Stiles, *Color Science: Concepts and Methods, Quantitative Data and Formulas,* second ed. New York: Wiley, 1982.

Graham D. Finlayson obtained the BSc degree in computer science from the University of Strathclyde (Glasgow, Scotland) and the MSc and PhD degrees in 1992 and 1995, respectively, from Simon Fraser University, Vancouver, British Columbia, Canada. From August 1995 to September 1997, he was a lecturer in computer science at the University of York, England and from October 1997 until August 1999, he was a reader in color imaging at the Colour and Imaging Institute, University of Derby, England. In September 1999, he was appointed a professor in the School of Information Systems, University of East Anglia, Norwich, England.

Steven D. Hordley obtained the BSc degree in mathematics from the University of Manchester, England in 1992. Then, he obtained the MSc degree in image processing from Cranfield Institute of Technology, Cranfield, England in 1996. Between 1996 and 1999, he studied at the University of York (York, England) and the University of Derby, England for the PhD degree in color science. In September 1999, he was appointed a senior research fellow at the University of East Anglia, Norwich, England and, in October 2000, he was appointed a lecturer at the same institution.

Paul M. Hubel received the BSc degree in optical engineering from The University of Rochester in 1986 and the DPhil degree from Oxford University in engineering science in 1990. His doctoral thesis was on color reflection holography. Dr. Hubel also worked as a postdoctoral researcher at MIT-Media Lab in the areas of color holography, color reproduction, and color holographic video. He has been a member of technical staff at Hewlett-Packard Laboratories for six years, working in the printing technology department. His recent work has been on illumination estimation and color correction for digital cameras; past projects include color reproduction for scanners, copiers, and printers. Dr. Hubel is a member of IS&T and SPIE.