A Re-evaluation of Colour Constancy Algorithm Performance

S. D. Hordley and G. D. Finlayson

School of Computing Sciences, University of East Anglia, Norwich NR4 7TJ, UK

This work is concerned with the evaluation of the relative performance of colour constancy algorithms. We highlight some problems with previous algorithm evaluation and define more appropriate testing procedures. We discuss how best to measure algorithm accuracy on a single image as well as suitable methods for summarising errors over a set of images. We also discuss how the relative performance of two or more algorithms should best be compared and we define an experimental framework for testing algorithms. We re-evaluate the performance of six colour constancy algorithms using the procedures we set out and show that this leads to a significant change in the conclusions we draw about relative algorithm performance as compared to previous work.

OCIS codes: 150.0150, 330.1690.

1. Introduction

An imaging system's response to light from an imaged scene depends on three factors: the underlying physical properties of the imaged surfaces, the nature of the light incident upon those surfaces and the characteristics of the imaging system itself. A fundamental problem in vision is that of disambiguating the effect of the scene illuminant on the recorded image from effects which are due to the underlying imaged surfaces. A successful solution to this problem is potentially useful for many visual tasks such as object recognition and tracking as well as for the more general problem of scene understanding. In addition, because our own visual system accounts (at least partially) for the colour of the light in a scene as part of its visual processing,[1–4] a solution to the problem is also important for image reproduction and digital photography where it is desirable for the colours in an image to be a good match to the scene as it was originally perceived by an observer.

Solutions to the *colour constancy problem* (as it is commonly called) usually proceed in two stages. First, an estimate of the colour of the scene illuminant is obtained from a recorded image and subsequently this estimate is used as the basis for a correction of the captured image to account for the effect of the prevailing scene illuminant. Usually this correction results in a re-rendering of the scene such that the newly rendered image corresponds to the scene as it would appear under some standard illumination. A great many colour constancy algorithms have been proposed (see for example[5–13] and[14] for a comprehensive review) in the literature with most attention being directed towards the first stage of colour constancy processing – estimating the scene illuminant from the recorded image data – since it is this stage which is most difficult to perform accurately. The theory of colour constancy processing is now quite well understood and there exist a number of sophisticated solutions to the problem. Nevertheless none of these solutions can be considered to be completely accurate and thus the colour constancy problem is still an active area of research.

In this paper we are concerned not with further advancing colour constancy theory but rather with an investigation into the performance of existing algorithms. The contributions of this work are threefold. First, we identify the weaknesses of previous evaluations of colour constancy algorithms. Second, we propose error metrics, statistical tests and an evaluation framework which we believe (for reasons justified in the paper) will enable an accurate assessment of existing and future algorithms. Finally, we re-evaluate a number of existing algorithms in the context of the framework proposed in this paper and show that this new evaluation leads to conclusions about the relative performance of the algorithms that are significantly different to the findings of previous studies.

It is usual in the literature, that when a new algorithm is proposed some kind of empirical evaluation of that solution is conducted. The extent of the evaluation however, varies from an informal demonstration of the algorithm's performance on a few images to a fuller evaluation on synthetic and/or real images. Even when evaluation is reasonably comprehensive however, the empirical framework usually differs from one work to another making a direct comparison of algorithm performance quite difficult. The work of Funt *et al*[15] addressed this shortcoming to some extent by investigating the performance of a number of colour constancy algorithms in a common experimental framework. Moreover, that work directly addressed the question as to whether or not the tested algorithms were good enough to allow a subsequent visual task to be successfully performed. Specifically, they investigated whether the performance of algorithms was sufficiently good to enable colour-content based object recognition in the context of a changing illumination. The authors concluded that the answer to this question was no, suggesting that further advancements in algorithm development are necessary before colour constancy algorithms are of practical use. Later work,[16] suggests that the probabilistic algorithm proposed in[13] does give good enough illuminant estimation accuracy to allow this task to be performed. However, the experiment was based on only a small number of images and it is therefore difficult to draw any definitive conclusions from it. In more recent work Barnard *et al*[17,18] investigated the performance of many colour constancy algorithms independently of any particular visual task concentrating instead on simply measuring the accuracy of the estimates of the scene illuminant provided by the algorithms in a set of experiments on both synthetic and real images.

The work of Barnard *et al* is the most comprehensive evaluation of colour constancy algorithms conducted to date and goes some way to achieving the goals of this paper. However, Barnard *et al*'s work (along with many other previous algorithm evaluations) is limited by the choice of error metric used to compare algorithms. It is common when comparing algorithm performance to look at the "average" performance of the tested methods over a set of images. Usually this "average" performance is reported in terms of a single summary statistic: for example the mean, or root mean square (RMS) error over the set. The fact that the chosen summary statistic is lower say, for algorithm A than for algorithm B, is used as a basis to conclude that algorithm A is "better" than algorithm B. Whether it is valid to draw such a conclusion depends in part on the underlying error distributions from which the summary statistics are calculated and also on the particular choice of summary statistic. We show in this paper that for the particular error distributions being studied the most commonly used statistics (mean or RMS error) do not give an accurate summary of the underlying distribution and thus any conclusions made on the basis of these statistics are suspect. We address this shortcoming by proposing a number of statistical measures of relative algorithm performance which are more appropriate to the data under investigation. We also discuss some additional measures of algorithm performance which provide more information than just a single summary statistic. Specifically, we discuss how to calculate confidence intervals for the summary statistics and how to assess the relative performance of algorithms in the context of these intervals. Further, we describe a number of appropriate hypothesis tests which can be used to determine the statistical significance of the differences between algorithms.

Having established the appropriate error measures by which to assess algorithm performance we turn our attention to the empirical framework in which algorithms are tested. Barnard *et al*'s evaluation framework is a good starting point. In particular, the fact that both the real and synthetic test data they used is publicly available[19] means that it is sensible to follow the experimental framework they proposed. To this end we repeat the experiments of Barnard *et al* on a subset of the algorithms they originally tested in the light of our investigation into how best to judge algorithm performance. We also suggest an alternative empirical framework for assessing algorithm performance on synthetic images which we believe addresses an important weakness of the original synthetic image experiment of Barnard *et al.* Our evaluation of six colour constancy algorithms on the three different experiments shows that the empirical framework and choice of error metric has a significant effect on the judgement of the relative performance of the algorithms. In the light of this re-evaluation we propose a set of guidelines for future algorithm evaluation.

The rest of the paper is organised as follows. In the next section we give a brief, formal introduction to the colour constancy problem which enables us to formulate some appropriate error measures for determining the accuracy of a given algorithm's estimate of the scene illuminant on a given image (Section 3). In Section 4 we look carefully at the most appropriate way to use these error metrics to evaluate the overall performance of an algorithm and to compare the relative performance of two or more different algorithms. Then, in Section 5 we describe the experiments we conducted to evaluate the performance of a number of different colour constancy algorithms. We give a brief overview of the tested algorithms, and evaluate their performance in Section 6. Finally, in Section 7 we conclude the paper by summarising a set of guidelines for evaluating colour constancy algorithms.

2. The Colour Constancy Problem

The colour constancy problem can be simply stated as the problem of how, given an image of a scene captured under an unknown illuminant, can we recover an estimate of that light? Solving the problem turns out to be difficult in practice and progress towards a solution cannot be made without more clearly defining the concept of what an image is and what we mean by an "estimate of the scene illuminant". An image usually consists of a 2-d array of sensor responses such that each element of the array represents an imaging device's response to light from a particular point in an imaged scene. To complete the definition of an image we must define the relationship between the light incident upon the imaging device and the device's response to that light. To this end it is common to adopt a simplified model of image formation in which the device's response to light from a point in the scene is given by:

$$p_k = \int_\omega S(\lambda)E^o(\lambda)Q_k(\lambda)d\lambda \tag{1}$$

where $S(\lambda)$ defines the surface characteristics at particular spatial location: it defines the proportion of light incident at that position which is reflected on a per-wavelength (λ) basis. $E^o(\lambda)$ is the spectral power distribution (SPD) of the scene illuminant; it characterises how much energy the source emits as a function of wavelength. $Q_k(\lambda)$ is the spectral sensitivity function of the imaging device sensor which determines what proportion of light energy incident upon it is absorbed at each wavelength. Thus, the sensor response p_k is a measure of the total energy absorbed by the sensor over the range of wavelengths ω to which the sensor is sensitive. The subscript k distinguishes a particular class of imaging sensor. In this paper we will concern ourselves with imaging devices which have (as is commonly the case) three different classes of sensor each with a different sensitivity profile. This implies that the response of an imaging device at a given point x is a triplet of sensor responses: $\underline{p} = [p_1 \;\; p_2 \;\; p_3]^t$. An image is then a collection such triplets which we can represent by the columns of a $3 \times N$ matrix P^o where the superscript o denotes the fact the image is captured under an unknown illuminant o.

Given this definition of an image formation we can define a general colour constancy algorithm as a function, denoted $\mathcal{C}(\cdot)$ which takes as its argument an image P^o and returns an estimate of the scene illuminant in P^o:

$$\hat{E}^o(\lambda) = \mathcal{C}(P^o) \tag{2}$$

This definition implies that $\mathcal{C}(\cdot)$ recovers an estimate of the scene illuminant, however in practice most algorithms recover only an estimate of the scene illuminant white-point: that is, the imaging device's response to a uniformly reflecting surface viewed under the scene illuminant:

$$\hat{\underline{p}}^o_w = \left[\hat{p}^o_{w,1} \;\; \hat{p}^o_{w,2} \;\; \hat{p}^o_{w,3}\right]^t = \mathcal{C}_w(P^o) \tag{3}$$

Other algorithms recover not a direct estimate of $\hat{\underline{p}}^o_w$ but rather return an estimate of a 3×3 diagonal matrix $\hat{D}^{c,o}$ which maps sensor responses under the scene light to their corresponding responses under a known, canonical or reference illuminant:

$$\underline{p}^c \approx \hat{D}^{c,o}\underline{p}^o \tag{4}$$

$$\hat{D}^{c,o} = \mathcal{C}_D\left(P^o, \; E^c(\lambda)\right) \tag{5}$$

In this case it is easy to derive an estimate of the scene illuminant by re-arranging Equations (4) and (5):

$$\hat{\underline{p}}^o = \left(\hat{D}^{c,o}\right)^{-1}\underline{p}^c = \left[\mathcal{C}_D\left(P^o, \; E^c(\lambda)\right)\right]^{-1}\underline{p}^c \tag{6}$$

Finally, other algorithms recover only an estimate of the scene illuminant's chromaticity: a 2-d intensity independent representation of the white-point:

$$\hat{\underline{c}}^o_w = \left[\hat{c}^o_{w,1} \;\; \hat{c}^o_{w,2}\right]^t = \mathcal{C}_c(P^o) \tag{7}$$

There are a number of ways of defining an intensity independent representation of an arbitrary sensor response $\underline{p}$. Throughout this paper we adopt the following:

$$c_1 = \frac{p_1}{p_1 + p_2 + p_3}, \;\; c_2 = \frac{p_2}{p_1 + p_2 + p_2}, \;\; c_3 = \frac{p_3}{p_1 + p_2 + p_3} \tag{8}$$

Note that since $c_1 + c_2 + c_3 = 1$ a chromaticity vector can be represented by any two of its elements e.g. $\underline{c} = \{c_1 \;\; c_2\}^t$. However, sometimes it is useful to represent this 2-d information in a 3-d form in which case we write:

$$\underline{q} = [c_1 \;\; c_2 \;\; (1 - c_1 - c_2)]^t \tag{9}$$

3. Measuring Algorithm Accuracy

Because different algorithms differ in their definition of "scene illuminant estimate" we must consider an appropriate way to compare different algorithms. In this paper we assess algorithms in terms of the accuracy of their estimate of the scene illuminant chromaticity since all algorithms either explicitly estimate this quantity or if not, such an estimate can easily be recovered from the quantity which is explicitly estimated. It might be argued that by restricting attention to the estimation of scene illuminant chromaticity we are ignoring an important aspect of algorithm performance since, for example, an algorithm which recovers an estimate of the intensity of the scene illuminant as well as its chromaticity provides a richer description of the scene illuminant. Similarly, if an algorithm estimates the SPD of the scene illuminant it seems in some sense unreasonable to ignore this fact in algorithm evaluation. However, in most situations accurate estimation of illuminant chromaticity is much more important than accurate estimation of its intensity. For example, to correct an image taken under an arbitrary scene illuminant to reference illumination conditions, it is sufficient to know the the chromaticities of the two illuminants since the overall intensity of the illumination is ambiguous. For this reason, and the fact that a chromaticity estimate is the only measure of algorithm performance common to all algorithms, we restrict ourselves to chromaticity error measures in this paper.

Two metrics are commonly used to quantify chromaticity error: the Euclidean distance between the 2-d chromaticity vectors and the angular distance between the 3-d representation (Equation (9)) of the two vectors. Euclidean distance is calculated:

$$e_{Euc} = \sqrt{(\hat{c}^o_{w,1} - c^o_{w,1})^2 + (\hat{c}^o_{w,2} - c^o_{w,2})^2} \tag{10}$$

and angular error is calculated:

$$e_{Ang} = \text{acos}\left(\frac{\underline{q}^{o\,t}_{w}\hat{\underline{q}}^o_w}{\|\underline{q}^o_w\|\|\hat{\underline{q}}^o_w\|}\right) \tag{11}$$

Typically, there is a high degree of correlation between these two error measures and so it is sufficient to evaluate performance using just one of them: we use angular error throughout this paper since it is perhaps more widely used in the literature.

Another factor to consider in algorithm evaluation is the colour space in which the error measures are calculated. Implicitly we have defined algorithm error in the space defined by the spectral sensitivity functions of the capture device. However, it might be considered that some other space would be more appropriate. For example, a number of (approximately) perceptually uniform spaces have been proposed[20] and measuring algorithm performance in such a space can in theory provide useful information. Whether this is in fact the case depends to a degree on the application for which the algorithm is being used. In computer vision applications for example, perceptual accuracy is not always relevant, whereas in digital photography a perceptual measure of accuracy is often important. However, to obtain a meaningful measure of perceptual accuracy we would have to take into account many more factors than simply the accuracy of the scene illuminant white-point. For example, factors such as the scene content and the conditions under which the image is viewed are very important. In addition, it is not clear that the aim in digital photography is to produce a rendering of the scene which is colorimetrically accurate since such a reproduction is not necessarily that which is preferred. Since addressing these issues is non-trivial (and indeed, an active area of research in itself) we believe that it is better not to confuse the issue by introducing measures of "perceptual accuracy" which may or may not be important.

Having said this, there are still advantages to be gained by looking at error in a space other than the sensor space of the imaging device. First, working in device space means that it is not possible to compare algorithm performance across different devices, a shortcoming that can be addressed by mapping sensor responses from an arbitrary imaging device into some standard colour space. Working in such a space also addresses the fact that the space defined by a device's sensor sensitivity curves can sometimes have quite unusual geometric properties with the implication that measuring error in the space is inappropriate. The disadvantage of mapping to a standard colour space is that it will not always be possible to map sensor responses to the standard colour space without error and so, we risk confounding the errors in an algorithm's illuminant estimate with the errors introduced by the colour space transformation. However, in most cases such transformations will map the illuminant white-point (and responses close to it) with good accuracy so that little error is introduced by the transformation. Moreover, what error is introduced, is introduced to all algorithms.

In the experiments reported in this paper we measure algorithm performance both in the colour space defined by the spectral sensitivities of the device under investigation, as well as in a more standard colour space: that defined by the XYZ colour matching functions.[20] We transform a response triplet $\underline{p}$ in device space to its corresponding triplet $\underline{x}$ in XYZ space by applying a 3×3 transform M:

$$\underline{x} = M\underline{p} \tag{12}$$

To obtain M we first calculate the sensor responses and their corresponding XYZ values for a representative set of surface reflectance functions (under a standard illuminant) according to Equation (1). We then determine the M which best transforms the XYZ values to their corresponding sensor responses using a standard least-squares approach.[21]

4. Evaluating and Comparing Algorithm Performance

The error measures we have introduced tell us the accuracy of a particular algorithm and allow us to easily compare the relative performance of two or more algorithms on a single image. Of course, algorithm performance will vary from image to image and so to obtain an accurate assessment of algorithm performance we must consider its performance over a large and diverse set of images.

When assessing algorithms it is common[13,14,17] for authors to summarise their performance in terms of their average performance over a large set of images using one (or a few) summary statistics. For example, the mean angular error or the Root Mean Square (RMS) chromaticity error over a set of images is quoted along with other summary statistics such as the maximum error. If the quoted statistic for algorithm A is found to be lower than that for algorithm B then the conclusion is drawn that algorithm A is better than algorithm B. There are two problems with this assessment. First, a single summary statistic such as the mean does not necessarily adequately summarise the underlying distribution. Second, the fact that one algorithm has a lower mean value than another is not sufficient information for drawing the conclusion that one algorithm is better than the other. More properly we can formulate a hypothesis that one algorithm is better than another and then test this hypothesis using appropriate statistical tools and the error distributions of each algorithm over a large set of sample images.

First, we consider the most appropriate summary statistic by which to compare algorithm performance. The most thorough evaluation of colour constancy algorithms to-date has been given by Barnard *et al.*[17,18] As part of their evaluation they looked at the distribution of chromaticity errors. That is, for a given algorithm they calculated:

$$e_{c1} = c^o_{w,1} - \hat{c}^o_{w,1}, \;\; e_{c2} = c^o_{w,2} - \hat{c}^o_{w,2} \tag{13}$$

for each image. They found the distribution of these errors to be approximately normally distributed with a mean of zero (the top left plot of Figure (1) illustrates that this is indeed the case). On this evidence they concluded that an appropriate error measure for assessing algorithm performance was the root mean square (RMS) error of a given error measure:

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N} {e_i}^2} \tag{14}$$

where N is the number of images over which the error is computed and e_i is the value of the particular error statistic being studied (e.g. angular error) for the i^{th} image. In the case that the chosen error measure is normally distributed with a mean of zero then RMSE gives an estimate of the standard deviation of the error statistic. However, the fact that $r^o_w - \hat{r}^o_w$ is normally distributed does not imply that other error measures are also normally distributed and in the event that they are not, RMS error is not necessarily an appropriate measure.

The top right plot of Figure 1 shows the distribution of chromaticity errors (calculated according to Equation (11) for a typical colour constancy algorithm (the *Max-RGB* algorithm) for 1000 images (generated using a procedure described in Section 5) each containing 8 surfaces. It is clear from this histogram plot that this error measure is not normally distributed. The bottom left plot in Figure 1 shows the distribution of angular errors for the same image set and once again, it is clear that the error measure does not follow a Normal distribution. This fact is emphasised by the bottom right plot which plots quantiles of a standard normal distribution against the quantiles of the angular error distribution for the 1000 images. If the errors were normally distributed the points on this plot would fall along a straight line. This example illustrates the typical case for the algorithms we have tested on both real and synthetic images. On this evidence we should conclude that angular error is not normally distributed so that RMS error does not give an estimate of the standard deviation of the error measure. So, if we want to look at a single summary statistic for this error distribution which should we choose? The mean error is often reported as a summary statistic however, it is well known[22] that the mean is a poor summary statistic for non-symmetric distributions: the distributions we are investigating are skewed as the example in Figure 1 illustrates. In these situations the median is a more reliable estimate of central tendency[22] thus we propose that if a single summary statistic is to be used to compare algorithms the median is the most appropriate measure.

A more informative summary of performance than a single summary statistic is to supplement it with a confidence interval for the statistic since this provides information about the likely variation in the statistic. In the case that the underlying error distributions are not well modelled by standard statistical distributions (as is the case here) care must be taken when calculating a confidence interval. To obtain confidence intervals for a given statistic in such a situation it is appropriate to use the method of re-sampling.[22] To understand this method let $\mathcal{E}$ represent the set of n error measurements for a particular algorithm obtained from a set of test images. Now, suppose we draw (with replacement) n samples from $\mathcal{E}$ to give us a new set of observations $\mathcal{E}_1$. We can calculate and record our chosen statistic, θ_1 for this new set of observations. We repeat this procedure a number (m) times where m is a large number. Each time we re-sample we obtain a new sample distribution ($\mathcal{E}_i$) whose statistic θ_i we can calculate. This provides us with a set of estimates of the chosen statistic (i.e. a distribution for θ). A $p\%$ confidence interval for the statistic can be obtained from the $p/2$ and $(1-p/2)$ quantiles of this distribution.

For example, using this re-sampling approach we can obtain confidence intervals for the median performance of two or more colour constancy algorithms which we can use to help us assess their relative performance. In the case that the two confidence intervals do not overlap at all we can draw the conclusion that there is a significant difference (at the $p\%$ significance level) between the two algorithms when judged according to the median statistic. If two confidence intervals overlap such that the mean (central point) of one or other interval falls within the second interval then we can conclude that there is no significant difference (again at the $p\%$ level) between the two algorithms. In the third case that the intervals overlap but the mean of neither interval lies within the second interval, we cannot draw a conclusion about the significance of the relative algorithm performance and we must resort to different methods.

To more formally determine the statistical significance of the differences between algorithm performance we should use hypothesis testing. When choosing an appropriate hypothesis test we must once again consider the underlying nature of the error distributions under study. In our case the error distributions are not well described by standard statistical distributions (e.g. a normal distribution) so commonly used statistical tests such as the Student's t-test are inappropriate. Instead, we should employ non-parametric tests which are independent of the underlying distribution. We consider two such tests here: the Sign Test[22] and the Kolmogorov-Smirnov (K-S) Test.[22] The Sign Test allows us to determine the significance of the difference between the median of two different distributions while the the K-S test is used to investigate the statistical significance of the differences between the distributions themselves.

Suppose that we wish to compare the relative performance of two algorithms in terms of their median angular error. We begin by using each algorithm to estimate the scene illuminant for a set of N images. Let A and B be random variables representing the error in algorithm A and B's estimate of the scene illuminant. The Sign Test can be used to test the hypothesis that the random variables A and B are such that $p = P(A > B) = 0.5$. That is we hypothesise that algorithm A and B have the same median:

$$H_0 : p = 0.5, \quad \text{the medians of the two distributions are the same} \tag{15}$$

We also define an alternative hypothesis:

$$H_1 : p < 0.5, \quad \text{algorithm A has a lower median than algorithm B.} \tag{16}$$

To test which of these hypotheses is true we consider independent pairs $(A_1, B_1) \ldots (A_N, B_N)$ of errors for N different images. We denote by W the number of images for which $A_i > B_i$. When H_0 is true W is binomially distributed ($b(N, 0.5)$) and the Sign Test is based on this statistic. Applying the Sign Test to our error data amounts to first determining W for a set of test images. Supposing that $W = w$ for a given set of test images, we next calculate $P(W \leq w)$, assuming that the null hypothesis is true. That is, assuming that $W \sim b(N, 0.5)$. Then, if $P(W \leq w) < \alpha$ we reject the null hypothesis H_0 and accept the alternative hypothesis H_1 at the significance level α. The value of α we choose determines the probability that we reject the null hypothesis when it is in fact true. So, for example if $\alpha = 0.05$ and the probability we calculate is 0.04 then we would reject the null hypothesis at the 0.05 significance level. In this case we will be correct in rejecting the null hypothesis

95% of the time. If we want to be more sure that we are correct we decrease the significance level.

While the Sign Test makes no assumption about the underlying distribution of the errors for each algorithm, in using this test we are making the implicit assumption that the median is a good summary statistic for the distributions. An alternative to single summary statistic comparisons of algorithms is to test whether or not the two error distributions themselves are significantly different. The K-S Test is applicable in this case. Like the Sign Test, the K-S Test makes no assumption about the underlying shape of the error distributions. The test statistic in this case is the maximum absolute difference between the two cumulative distributions:

$$D = \max_{-\infty < x < \infty} |C_A(x) - C_B(x)| \tag{17}$$

where $C_A(x)$ and $C_B(x)$ are the cumulative distributions corresponding to the two error distributions under investigation. To apply this test we define the null hypothesis

$$H_0 : C_A(x) = C_B(x), \quad \text{the two distributions are the same} \tag{18}$$

We also define an alternative hypothesis:

$$H_1 : C_A(x) < C_B(x), \quad \text{Errors for algorithm A are lower than those for algorithm B} \tag{19}$$

which if true implies that algorithm A performs better than algorithm B. For a given pair of cumulative distribution functions $C_A(x)$ and $C_B(x)$ and under the assumption of H_0, we can calculate[23] the probability that D has a probability greater than that observed using:

$$P(D > \text{observed}) = Q_{KS}\left(\left[\sqrt{N_c} + 0.12 + 0.11/\sqrt{N_c}\right]\right) \tag{20}$$

where

$$Q_{KS}(y) = 2\sum_{j=1}^{\infty}(-1)^{j-1}e^{-2j^2y^2} \tag{21}$$

and $N_c = N^2/(2N)$. If the probability we calculate is less than the chosen significance level α, we reject the null hypothesis that the distributions are the same. As in the case of the Sign Test the significance level α determines the probability that we are wrong in this rejection.

5. An Experimental Framework for Algorithm Evaluation

Having established appropriate methods for determining the relative performance of two or more colour constancy algorithms we turn our attention to defining an appropriate experimental framework in which to evaluate algorithms. Ideally all algorithm performance would be conducted using images captured with real imaging devices since this provides a true test of how algorithms perform. However, many algorithms work under the assumption that they have some prior knowledge about the characteristics of the imaging device (e.g. its spectral sensitivity functions) so that compiling a useful test set of real images is a non-trivial exercise. This task is made more difficult due to the fact that for an image to be useful in testing algorithms, we must have accurate knowledge of the scene illuminant. A simpler approach is to test algorithms on images synthesised for a hypothetical imaging device. This makes it easy to assess algorithms using many different images containing a wide range of surfaces and captured under many different lights. The disadvantage of synthetic image tests is the fact that artefacts of the imaging process (e.g. image noise) are excluded so that we get a best-case assessment of algorithm performance. Here we test algorithms both on synthetic images and on a set of well calibrated real test images[18] captured explicitly for the purpose of evaluating colour constancy algorithms.

A. Synthetic Image Experiments

For our synthetic image experiments we generate sensor responses according to a simple Lambertian model of image formation:[24]

$$\begin{array}{l} p_1^o = \sum_{j=1}^{M} e^o(\lambda_j)s(\lambda_j)q_1(\lambda_j) \\ p_2^o = \sum_{j=1}^{M} e^o(\lambda_j)s(\lambda_j)q_2(\lambda_j) \\ p_3^o = \sum_{j=1}^{M} e^o(\lambda_j)s(\lambda_j)q_3(\lambda_j) \end{array} \tag{22}$$

where $e^o(\lambda_j)$, $s(\lambda_j)$ and $q_?(\lambda_j)$ represent M sample, discrete representations of an illuminant SPD, surface reflectance and sensor sensitivity functions respectively. An image is then just a collection of n such sensor response triplets and by varying n we can investigate algorithm performance as a function of the number of distinct surfaces in an image.

Barnard *et al*[17] conducted a synthetic image experiment in which images were created using reflectances randomly selected from a set of 1995 measured reflectances. For each image a single scene illuminant was randomly selected from a set of 287 measured illuminants and sensor responses were created according to Equation (22) using a set of spectral sensitivity functions from a Sony DXC-900 video camera. Algorithm performance was evaluated for images with number of surfaces $n = 2$, 4, 8, 16, 32, or 64 and for each value of n 1000 images were generated. A number of algorithms tested in that experiment require what amounts to a training phase in which they make use of information about which surface reflectance functions and/or illuminants occur in the world. Where an algorithm requires information about surface reflectance functions Barnard *et al* used the set of 1995 surface reflectances from which the synthetic images were constructed. In the case that an algorithm required information about possible scene illuminants a subset of 87 of the 287 scene lights were selected as possible scene illuminants. These 87 illuminants were chosen such that their chromaticities represented an approximate uniform sampling of the region of chromaticity space represented by the larger set of lights.

Providing an algorithm with completely accurate information about the lights and surfaces it will encounter ensures that we will obtain the best possible performance from any given algorithm. However, in practice it is difficult to ensure that, for example, the surfaces on which an algorithm is trained exactly match those in the images on which it is tested. For this reason we also carried out a modified form of Barnard *et al*'s synthetic image experiment in which we trained algorithms on a different set of surfaces to those in the images on which the algorithms were tested. Specifically we divided the set of 1995 reflectances into 5 equal sized random subsets. We then trained each algorithm on surfaces from 4 of the 5 subsets before testing them on images synthesised from the fifth surface set. We repeated this procedure 5 times: each time withholding a different subset from the training set. In this way we are able to investigate the robustness of algorithms to a mismatch between their training and testing conditions. In this second experiment we selected training illuminants exactly as described for the first experiment. When selecting training illuminants we could have adopted a procedure similar to our method of selecting training surfaces. However, we argue that it is easier to ensure that we select an appropriate gamut of possible scene illuminants than it is to ensure that we train on a set of surfaces which reflects the gamut and relative frequency of occurrence of surfaces in the world. Thus an analysis of the effects of a mismatch between training and testing illuminants is not so interesting. Of course, how finely we sample the gamut of illuminants when training an algorithm will affect the accuracy of any given algorithm's performance. However, in practice we can choose sufficient illuminants to give us any level of accuracy we choose so provided we select the same set of possible illuminants for all algorithms, the relative performance of algorithms is likely to be unaffected by our choice.

B. *Real Image Experiments*

For our experiments with real images we again followed the paradigm of Barnard *et al.*[18] We used images of 32 different scenes captured under up to 11 different lights (most scenes were captured under all 11 lights and a total of 321 images were used in this experiment). A measurement of the actual white point of the scene illuminant was obtained for each image by placing a white tile centrally in a scene and imaging it a second time. The camera's *RGB* response to this white tile is used as the actual white point for a given scene. Each algorithm returns an estimate of the scene illuminant whose accuracy we assess as described above. Where an algorithm requires a training phase we used training parameters exactly as described for the first synthetic image experiment.

6. Empirical Evaluation

In this section we evaluate the performance of six different colour constancy algorithms. We begin with a brief description of each algorithm and then we present the results of their performance in the experiments on synthetic and real images.

A. *Algorithms Tested*

The six algorithms tested are *Max-RGB*,[5] Grey world (denoted *GW*),[6,20] Database Greyworld[7] (denoted *DB GW*), a version of the Gamut Mapping[12,14,25] algorithm (denoted *LP GM*), a version of Colour by Correlation[13] (denoted *CbyC*) and a Neural Network algorithm[26] (denoted *NN*). These six algorithms are representative of the state-of-the-art in colour constancy algorithms and include the best performing algorithms according to the evaluation reported by Barnard *et al.*[17,18]

The Max-RGB algorithm returns an estimate of $\underline{p}_w^o$: the scene illuminant white point. This estimate is found simply by calculating the maximum sensor response in each channel of an image. We can expect this algorithm to work well when an image contains a white surface or surfaces which are maximally reflective in the red, green and blue regions of the spectrum.

The Grey world algorithm also returns an estimate of $\underline{p}_w^o$. This algorithm is founded on the assumption that the average of all surface reflectances in an image is neutral (grey) which implies that an estimate of the scene illuminant can be found by calculating the mean sensor response for each channel of an image. The Database Grey world algorithm is similar except that the average of all surfaces in the the image is assumed to correspond to the average of a pre-compiled set of reflectances rather than a neutral reflectance. In this case the estimate of the scene illuminant white point is given by:

$$\hat{p}_{w,k}^o = p_{w,k}^c \frac{p_{m,k}^o}{p_{m,k}^c} \quad k = 1, 2, 3 \tag{23}$$

where $\underline{p}_m^o$ is the mean image response and $\underline{p}_m^c$ is the mean response to all possible surface reflectances when viewed under a known reference light c.

Gamut mapping algorithms were first introduced by Forsyth.[12] In this algorithm a canonical gamut is first defined: it is the set of all sensor responses observable under a (known) reference light. Similarly, an image whose illuminant is to be estimated is represented by the gamut of its sensor responses. In gamut mapping the aim is to find the diagonal mapping which takes the image gamut into the canonical gamut. In practice, for a given image, there will exist many different diagonal mappings which map the image gamut to the canonical gamut. Gamut mapping solutions first determine (implicitly or explicitly) the set of all consistent mappings and then apply an appropriate selection criterion to determine a single mapping as the scene illuminant estimate. Given this estimate we can obtain an estimate of the scene illuminant white point using Equation (6). There are different ways to implement the gamut mapping algorithm: here we use a linear programming implementation in which we find the diagonal mapping (d_1, d_2, d_3) with maximum sum subject to a set of linear constraints which are derived from the canonical gamut and the image RGBs and which ensure that the recovered mapping takes the image gamut into the reference gamut. This implementation has the advantage of being simple to implement and its performance has been shown[25] to be very similar to that tested by Barnard *et al.*[17]

The Color by Correlation algorithm was introduced by Finlayson *et al.*[13] In this algorithm the set of plausible scene illuminants is defined *a priori*. Each plausible light is characterised by a 2-d chromaticity distribution which gives a measure of the likelihood of observing any given chromaticity value under the light in question. An estimate of the scene illuminant white point in a particular image is found by first determining which chromaticities are present in the image. Each image chromaticity has a certain likelihood (recorded in the chromaticity distributions) of being observed under each of the plausible lights and the sum of the likelihood values for all image chromaticities defines the likelihood that a given plausible light is the scene light. The plausible light whose likelihood value (summed over all image chromaticities) is highest is chosen as the scene illuminant. Because plausible lights are represented by 2-d chromaticity distributions *CbyC* can only recover a 2-d estimate of the scene illuminant (Note: an implementation of the Color by Correlation idea in a 3-d colour space has been proposed[27] but is not tested in this work). This estimate takes the form of the chromaticity of the scene illuminant white point. Performance of the algorithm depends strongly on how the chromaticity distributions for each plausible light are determined. In this work we followed the method reported in[17] as closely as possible. Distributions are defined in the chromaticity space defined by Equation (8) which is uniformly partitioned into 50×50 bins. A histogram of the chromaticities of a reference set of surface reflectances is obtained and this histogram is then smoothed using convolution with a Gaussian kernel to obtain the final chromaticity distribution.

The final algorithm tested uses a neural network to estimate the scene illuminant. Input to the network takes the form of a binary chromaticity histogram: a vector of ones and zeros such that each element of the vector corresponds to a small sub-region of a 2-d chromaticity space. A value of one in a particular element of the vector implies that a chromaticity value falling in the corresponding sub-region of chromaticity space was found in the image, otherwise the vector element is zero. The output of the neural network is a 2-d chromaticity value: an estimate of the chromaticity of the scene illuminant white point. The particular neural network used in these experiments is as close as possible to the best performing networks reported in.[26] The chromaticity space used to define the input layer (the binary histogram) is that defined in Equation (8) and it is uniformly partitioned into

50 × 50 bins. The network has two hidden layers (with 400 and 30 neurons respectively) and is trained using back-propagation. Training was performed using synthetic images and following (as far as possible) the scheme suggested in.[26] We note however, that the training data we used in this paper is quite different to that used in.[26] This is deliberate since we wanted to train all algorithms on the same data. This difference in training data probably explains, in large part, the difference in the results we obtain for this algorithm compared to those reported in[26] and.[17]

B. *Experiment 1 Results*

In their original experiment Barnard *et al* summarised algorithm performance using the RMS error measure (Equation (14)) calculated over all images with a given number of surfaces. Figure 2 summarises the performance of the six algorithms tested in this work using the RMS measure. As we would expect algorithm performance is dependent on the number of surfaces in an image and the performance of all algorithms improves as the number of surfaces is increased. In addition, the relative performance of different algorithms changes as the number of surfaces in an image changes. However, RMS angular error gives only one view of algorithm performance and, for the reasons we discussed in Section 4, it is not the most appropriate statistic by which to judge the algorithms. Figure 3 summarises algorithm performance (as a function of number of surfaces per image) in terms of the median angular error statistic. Importantly Figure 2 gives quite a different view of relative algorithm performance compared to Figure 1: the relative rank ordering of algorithms is quite different in the two figures. This fact is better illustrated by Table 1 which shows the rankings of the different algorithms (as a function of the number of surfaces in an image) both in terms of RMS and median angular error. Comparing columns 1-6 of this table with columns 7-12 we see that the rankings of the six algorithms varies considerably depending on whether we use RMS or median error. While the rank of most algorithms changes by only one or two positions this observation is nevertheless significant and highlights the importance of the choice of error measure by which to compare algorithms. Columns 1-2 of Table 2 summarise the overall rankings of the six algorithms judged according to RMS and median angular error. In this case the statistics and subsequent rankings are calculated over all 6000 test images. The position of only one algorithm (CbyC) is unchanged depending on whether we use RMS or median error while all other algorithms change their position by at least one place. Column 3 of this table ranks algorithms according to their mean angular error and once again a different ordering results. Note that, like RMS, mean error is inappropriate in this case since it is a poor measure of central tendency for skewed distributions.

Having established that the choice of summary statistic has a significant effect on our judgement of relative algorithm performance we next consider the statistical significance of the results. Figure 4 goes some way to establishing this: here we have plotted median angular error for the best three performing algorithms (judged according to the overall median error) as a function of number of surfaces per image but in addition we have plotted 95% confidence intervals for each point. It is clear from this figure that for some cases there is a significant difference between algorithms. For example, for images containing up to 8 surfaces the difference between *CbyC* and both *DB GW* and *LP GM* is significant (at the 95% significance level) while for images with more than this number of surfaces there is no significant difference until we reach 64 surfaces where *DB GW* and *LP GM* are significantly better than *CbyC*.

To properly assess the significance of the differences in performance we resort to hypothesis testing. Columns 5 and 6 of Table 2 rank the six algorithms according to the Sign Test and the Kolmogorov-Smirnov Test (at a significance level of 0.01) over all 6000 test images. The Sign Test is used to determine whether or not the difference between the median result for each algorithm is different: the results show that the differences are significant except between algorithm *DB GW* and *LP GM* which have a tied rank. The Kolmogorov-Smirnov test judges algorithms according to the differences between their error distributions rather than based on a single summary statistic. In this case the ranking of algorithms according to this test is identical to that obtained with the Sign Test (which in turn is very similar to that obtained based on just the median statistic). This result suggests that if algorithms are to be judged on just a single summary statistic then the median is the appropriate one to use.

Next, we turn our attention to the space in which algorithm error is measured. In Section 4 we proposed that for a number of reasons it is a good idea to measure algorithm error in a standard colour space rather than the space defined by the device being studied. To this end, Table 3 summarises the relative overall performance of the six algorithms in terms of angular error measured in the space defined by the CIE 1931 Colour Matching Functions[20] (CMF). To map illuminant estimates to the CMF space we followed the procedure outlined in Section 3 above deriving a least-squares transform based on the 1995 surfaces used to create the synthetic images. In this experiment the choice of colour space makes little difference: the overall rankings of the algorithms judged according to four out of the five methods is unchanged by a change of colour space. Interestingly, the only change in rank ordering occurs when we judge algorithms in terms of RMS error.

Finally we look at algorithm performance in terms of chromaticity error rather than angular error. Again we measure chromaticity error in the space defined by the Colour Matching Functions and Table 4 summarises the relative performance of the six algorithms according to the different summary statistics and statistical tests. Since the two error measures are in some sense measuring the same thing we would expect that differences between the two measures would be minor. Table 4 suggests that this is indeed the case: some small differences are introduced by the change of error measure but on the whole the changes are minor.

C. Experiment 2 Results

In the second synthetic image experiment we evaluated algorithm performance when the training data with which algorithms were provided differed from the data used to create the synthetic images. If the algorithms are robust to differences between training and testing data we would expect to see similar performance in this experiment as we saw in the first experiment. Figure 5 plots the median angular error for the six algorithms as a function of the number of surfaces per image. In this experiment algorithms were trained on five different data sets and five sets of results are thus obtained. Figure 5 shows the median error calculated over all five trials. Algorithm performance follows the same trends as the first experiment with the performance of all algorithms (save Grey World) converging as the number of surfaces increases. The errors for all algorithms are increased, a fact borne out by Table 6 which shows the overall median errors for the six algorithms in the first and second experiment. Hypothesis testing (the KS Test at a significance level of 0.01) reveals that in all cases the difference in algorithm performance is significant. That is, the performance of all algorithms is worse in this experiment. We might expect this for four of the six algorithms since the performance is tied to some extent to the quality of the training data. However, the remaining two algorithms (Max-RGB and Grey World) also perform worse despite the fact that they do not require training. Their worsening performance can therefore only be caused by the fact that the reflectances used to synthesis images are changed. This highlights the fact that algorithm performance can be affected by the choice of test data and confirms the need for a careful testing procedure. In this case, the test data used in the first experiment produces an optimistic view of the performance of these two algorithms.

Table 7 summarises algorithm rankings in this second experiment. Once again we note that algorithm ranking depends on the choice of error metric. Overall rankings for the six algorithms are largely unchanged in the two experiments from which we conclude that while the performance of all algorithms is worse, no one algorithm is more significantly affected than another. The exception to this being Grey World which was anyway the worst performing algorithm. Those algorithms which include a training step appear to be equally robust to differences between the training and testing data. The experiments on real image data will provide further information as to the extent to which this statement is true.

D. Experiment 3 (Real Image) Results

In a final experiment we evaluated algorithm performance on real images. As noted by Barnard *et al* we found that algorithm performance on real images varies considerably depending on the pre-processing we apply to an image prior to estimating the illuminant. In general algorithm performance is improved either by segmenting or down-sampling images prior to the estimation step. Here, for four of the six algorithms (Max-RGB, GW, $DBGW$ and NN) we used the segmentation procedure suggested by Barnard *et al.*[17] For $CbyC$ we used this same segmentation procedure but in addition we used only bright pixels (pixels with a brightness greater than the $70th$ quantile) as input to the algorithm. Finally for $LPGM$ we found that simply down-sampling images gave better performance than a segmentation approach.

Table 8 summarises the results of the real image experiments using RMS, mean and median error calculated over the 321 real images. It is clear from this table that results are again highly dependent on the choice of error metric used to evaluate algorithms. For example $CbyC$ is only the third best algorithm if judged according to RMS error but is ranked first according to median error. Most algorithms have a level of performance similar to that obtained for synthetic images with between 8 and 16 surfaces. Table 10 summarises the rankings of the algorithms using the three error measures as well as the Sign Test and KS Test. The overall trends are similar to those obtained for the synthetic images except for two main differences. First, $Max - RGB$ performs better on the real images than it does in the synthetic experiments and second, the difference between $CbyC$ and $LPGM$ appears to be less significant than in the synthetic image experiments: the algorithms are statistically equivalent according to the hypothesis tests on the real image results. Note also that while the median error for $CbyC$ is the lowest its RMS error is quite high. This suggests that for this algorithm there are some outliers: i.e. images for which the algorithm performs very poorly (the RMS error measure is more sensitive to outliers than the median). This highlights the fact that while the median is generally a more appropriate single summary statistic than RMS error, it does not imply that RMS error has no value. For example, in some applications it may be that an algorithm which works quite well for all images and which has no (or very few) extreme failure cases, is more useful than one that has a very low average error but which gives big errors on a few images. RMS error helps to identify these classes of algorithm.

Tables 9 and 11 summarise performance in XYZ space rather than device RGB space. Again, the overall trends in performance are similar but in this case the rankings based on the hypothesis tests show that the difference between algorithms is not always significant. For example, according to the KS Test algorithms can be grouped into three subsets: $CbyC$ and $LPGM$ perform similarly followed by $Max - RGB$ which performs better than three remaining algorithms whose performance is again similar. The difference between the results when error is judged in XYZ rather than sensor space again highlights the importance of choosing an appropriate space in which to assess performance. In this case results are similar so we might conclude that either space is appropriate, however, working in XYZ space leads to more conservative conclusions about the relative difference in performance between algorithms.

7. Conclusions

It is clear from the results of the experiments presented above that the relative performance of colour constancy algorithms is strongly affected by the choice of error metrics used to compare them. We conclude this paper by summarising the important issues that need to be considered in any future assessment of algorithm performance.

(1) The choice of error metric should be governed in part by the actual quantity estimated by the tested algorithms. In general however, error measures such as the angular error between estimated and actual white point are preferable to "perceptual errors" (e.g. CIELab error) because there are more factors than just the white point which affect the perception of an image.

(2) Thought should be given to the choice of colour space in which errors are measured. In particular, measuring error in XYZ space rather than device space should be considered since this makes it easier to judge the relative performance of different algorithms when they are tested on different devices and also means that errors are unbiased by the peculiarities of a particular imaging device.

(3) When summarising algorithm performance over a set of images thought should be given as to the most appropriate choice of summary statistic. The experiments in this paper suggest that the median statistic is more appropriate than the commonly used RMS or mean error. However, in general a suitable summary statistic should be chosen on the basis of the underlying error distributions being studied.

(4) Any single summary statistic can be made more informative by supplementing it with a confidence interval. A confidence interval for any statistic can be calculated using the re-sampling approach described above which is independent of the underlying error distribution.

(5) Appropriate hypothesis tests should be conducted to evaluate the statistical significance of differences in algorithm performance. The most appropriate hypothesis test depends in part on the choice of summary statistic. For example the Sign Test can be used to determine the significance of differences in the median of two distributions. The use of hypothesis tests such as the KS Test which investigate the significance of the difference between the two distributions themselves should also be considered, particularly when the underlying distributions cannot be well approximated by standard distributions.

(6) Evaluation of algorithm performance should, as far as possible, follow a well established experimental paradigm so as to make it easy to compare studies in different works. The synthetic and real image experiments reported by Barnard *et al*[17,18] together with the additional experiments reported in this work are an appropriate starting point.

We can also make some concluding remarks regarding the current state-of-the-art in algorithm performance based on the work presented in this paper.

(1) Two algorithms: *CbyC* and *LPGM* give significantly better performance than all other algorithms tested in both synthetic and real image experiments. The synthetic image experiments suggest that *CbyC* is capable of a higher level of performance than *LPGM* but real image tests suggest that their performance is very similar. The Grey World algorithms are the least successful algorithms and on real images give a very poor level of performance. Max-RGB performs well in the real image experiments: better than might be expected from the synthetic image results and given the simplicity of the method. Further testing on real images is required to ascertain whether the level of performance of Max-RGB in these experiments represents its true level.

(2) Many algorithms require a training phase prior to the estimation of the scene illuminant and it is likely that changes in this training procedure will have a significant effect on real image performance. In particular algorithms such as the neural network, *CbyC* and *LPGM* are sensitive to the training procedure and the performance of algorithms can likely be improved by choosing training data which better reflects the data encountered in real images.

(3) The synthetic and real image results follow similar trends implying that a good idea of algorithm performance can be obtained using synthetic images. However larger databases of real images are required to obtain a truer assessment of current algorithm performance on real images.

(4) Algorithm performance on real images is significantly affected by data pre-processing. Further research is required to determine the most appropriate pre-processing for each algorithm.

Author Contact Information

S. D. Hordley,
School of Computing Sciences,
University of East Anglia,
Norwich,
NR4 7TJ
UK

e-mail: steve@cmp.uea.ac.uk

G. D. Finlayson,
School of Computing Sciences,
University of East Anglia,
Norwich,
NR4 7TJ
UK

e-mail: graham@cmp.uea.ac.uk

References

1. David H. Brainard, Wendy A. Brunt, and Jon M. Speigle, ”Color constancy in the nearly natural image. I. Asymmetric matches,” Journal of the Optical Society of America, A, **14** 2091–2110 (1997).
2. David H. Brainard, ”Color constancy in the nearly natural image. 2. Achromatic loci,” Journal of the Optical Society of America, A, **15** 307–325 (1998).
3. Marcel Lucassen, *Quantitative Studies of Color Constancy* PhD thesis, (Utrecht University, 1993).
4. Lawrence Arend and Adam Reeves, ”Simultaneous color constancy,” Journal of the Optical Society of America, A, **3** 1743–1751, (1986).
5. Edwin H. Land, ”The Retinex Theory of Color Vision,” Scientific American, 108–129 (1977).
6. G. Buchsbaum, ”A spatial processor model for object colour perception,” Journal of the Franklin Institute, **310** 1–26 (1980).
7. Ron Gershon, Allan D. Jepson, and John K. Tsotsos, ”From [R,G,B] to Surface Reflectance: Computing Color Constant Descriptors in Images,” Perception, 755–758 (1988).
8. Laurence T. Maloney and Brian A. Wandell, ”Color constancy: a method for recovering surface spectral reflectance,” Journal of the Optical Society of America, A, **3** 29–33 (1986).
9. S. A. Shafer, ”Using color to separate reflection components,” Color Research and Application, **10** 210–218 (1985).
10. Hsien-Che Lee, ”Method for computing scene-illuminant chromaticity from specular highlights,” in *Color*, Glenn E. Healey and Steven A. Shafer and Lawrence B. Wolff eds. (Jones and Bartlett, Boston, 1992), pp. 340–347.
11. Glenn Healey, ”Estimating spectral reflectance using highlights,” in *Color*, Glenn E. Healey and Steven A. Shafer and Lawrence B. Wolff eds. (Jones and Bartlett, Boston, 1992), pp. 335-339.
12. D. A. Forsyth, ”A Novel Algorithm for Colour Constancy,” International Journal of Computer Vision, **5** 5–36 (1990).
13. G.D. Finlayson, S.D. Hordley, and P.M. Hubel, ”Color by correlation: A simple, unifying framework for color constancy,” IEEE Transactions on Pattern Analysis and Machine Intelligence, **23** 1209–1221 (2001).
14. K. Barnard, *Practical Colour Constancy.* PhD thesis, (Simon Fraser Univ., School of Computing Science, 2000).
15. Brian Funt, Kobus Barnard, and Lindsay Martin, ”Is machine colour constancy good enough?,” in *Proceedings of 5th European Conference on Computer Vision*, (Springer, 1998) pp. 455–459.
16. Graham. D. Finlayson, Steven Hordley, and Paul Hubel, ”Illuminant estimation for object recognition,” COLOR research and application, 260–270 (2002).
17. Kobus Barnard, Vlad Cardei, and Brian Funt, ”A comparison of computational color constancy algorithms; part one: Methodology and experiments with synthetic images,” IEEE Transactions on Image Processing, **11** 972–984 (2002).
18. Kobus Barnard, Lindsay Martin, Adam Coath, and Brian Funt, ”A comparison of computational color constancy algorithms; part two: Experiments with image data,” IEEE Transactions on Image Processing, **11** 985–996 (2002).
19. `http://www.cs.sfu.ca/∼colour/data/colour_constancy_test_images/index.html`.

20. R.W.G. Hunt, *The Reproduction of Colour*, (Fountain Press, 5th edition, 1995).
21. Gilbert Strang, *Linear Algebra and its Applications*, (Saunders College Publishing, 1988).
22. Robert V. Hogg and Elliot A. Tanis, *Probability and Statistical Inference*, (Prentice Hall, 2001).
23. William H. Press, Saul A. Teukolsky, William T. Vetterling and Brian P. Flannery, *Numerical Recipes in C The Art of Scientific Computing*, (Cambridge University Press, 1992).
24. Berthold K. P. Horn, *Robot Vision*, (MIT Press, 1986).
25. Graham D. Finlayson and Ruixia Xu. Convex programming colour constancy. in *Proceedings of Workshop on Color and Photometric Methods in Computer Vision*, (IEEE, 2003).
26. Vlad C. Cardei, Brian Funt, and Kobus Barnard, "Estimating the scene illuminant chromaticity by using a neural network," Journal of the Optical Society of America, A, **19** 2374–2386, (2002).
27. Kobus Barnard, Lindsay Martin and Brian Funt, "Colour by correlation in a three dimensional colour space," in *Proceedings of the 6th European Conference on Computer Vision*, 275-289, (2000).

Table 1. Experiment 1. Rankings by number of surfaces per image using RMS and Median Angular Error in Sensor Space.

	RMSE						Median					
Num. Surf	2	4	8	16	32	64	2	4	8	16	32	64
Mx-RGB	6	6	6	5	5	4	6	6	6	5	5	4
GW	5	5	5	6	6	6	3	5	5	6	6	6
DB GW	3	2	2	2	1	1	4	2	3	3	2	1
NN	2	3	4	4	4	5	2	4	4	4	4	5
LP GM	4	4	3	3	2	2	5	3	2	1	1	2
CbyC	1	1	1	1	3	3	1	1	1	2	3	3

Table 2. Experiment 1. Rankings over all 6000 images based on Angular Error in Sensor Space.

Algorithm	RMS	Median	Mean	Sign Test	KS Test
MxRGB	6	5	5	5	5
GW	5	6	6	6	6
DB GW	2	3	2	2	2
LP GM	4	2	3	2	2
CbyC	1	1	1	1	1
NN	3	4	4	4	4

Table 3. Experiment 1. Rankings over all 6000 images based on Angular Error in XYZ Space.

Algorithm	RMS	Median	Mean	Sign Test	KS Test
MxRGB	6	5	5	5	4
GW	4	6	6	6	6
DB GW	2	3	2	2	3
LP GM	3	2	3	2	2
CbyC	1	1	1	1	1
NN	5	4	4	4	4

Table 4. Experiment 1. Rankings over all 6000 images based on Chromaticity Error in XYZ Space.

Algorithm	RMS	Median	Mean	Sign Test	KS Test
MxRGB	6	4	5	5	5
GW	5	6	6	6	6
DB GW	2	3	2	2	2
LP GM	3	2	3	2	3
CbyC	1	1	1	1	1
NN	4	5	4	4	4

Table 5. RMS, median, and mean angular error computed in XYZ Space over the 321 real images.

Algorithm	RMS Error	Median Error	Mean Error
MxRGB	5.95	3.03	4.33
GW	9.07	5.90	7.21
DB GW	8.44	4.87	6.48
NN	7.96	5.14	6.34
LP GM	5.07	2.30	3.54
CbyC	7.72	2.16	4.76

Table 6. Overall Median angular error for the six algorithms in the two synthetic image experiments

Algorithm	Experiment 1	Experiment 2
MxRGB	4.47	5.23
GW	5.63	7.65
DB GW	3.85	4.42
NN	4.45	5.02
LP GM	3.78	4.37
CbyC	3.55	4.00

List of Figure Captions

Table 7. Experiment 2. Rankings over all 6000 images based on Angular Error in Sensor Space.

Algorithm	RMS	Median	Mean	Sign Test	KS Test
MxRGB	6	5	5	5	5
GW	5	6	6	6	6
DB GW	2	3	2	2	2
LP GM	4	2	3	2	3
CbyC	1	1	1	1	1
NN	3	4	4	4	4

Table 8. RMS, median, and mean angular error computed in Sensor Space over the 321 real images.

Algorithm	RMS Error	Median Error	Mean Error
MxRGB	8.88	4.05	6.38
GW	14.52	8.94	11.48
DB GW	12.44	6.85	9.44
NN	11.04	7.78	9.18
LP GM	6.85	3.71	5.00
CbyC	10.09	3.19	6.56

Table 9. RMS, median, and mean angular error computed in XYZ Space over the 321 real images.

Algorithm	RMS Error	Median Error	Mean Error
MxRGB	5.95	3.03	4.33
GW	9.07	5.90	7.21
DB GW	8.44	4.87	6.48
NN	7.96	5.14	6.34
LP GM	5.07	2.30	3.54
CbyC	7.72	2.16	4.76

previous plot.

Fig. 2. RMS angular error for each of the six algorithms tested in Experiment 1 as a function of ($\log_2$) number of surfaces in an image.

Table 10. Experiment 1. Rankings over all 321 real images based on Angular Error in Sensor Space.

Algorithm	RMS	Median	Mean	Sign Test	KS Test
MxRGB	2	3	2	3	3
GW	6	6	6	6	6
DB GW	5	4	5	4	4
LP GM	1	2	1	1	1
CbyC	3	1	3	1	1
NN	4	5	4	4	5

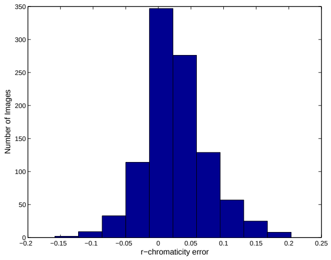
Table 11. Experiment 1. Rankings over all 321 real images based on Angular Error in XYZ Space.

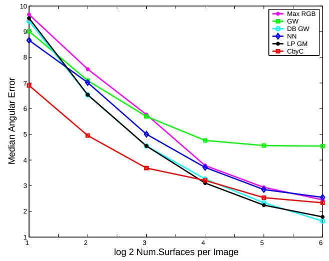
Algorithm	RMS	Median	Mean	Sign Test	KS Test
MxRGB	2	3	2	3	3
GW	6	6	6	6	4
DB GW	5	4	5	4	4
LP GM	1	2	1	1	1
CbyC	3	1	3	1	1
NN	4	5	4	4	4

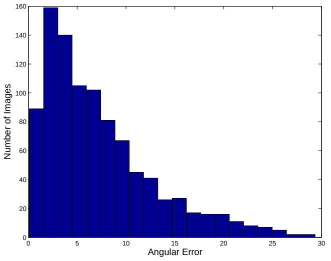
Fig. 3. Median angular error for each of the six algorithms tested in Experiment 1 as a function of ($\log_2$) number of surfaces in an image.

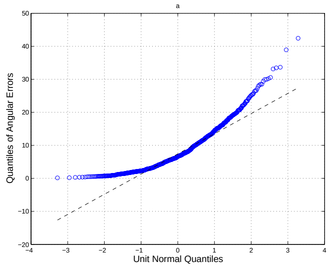
Fig. 4. Median angular error together with 95% Confidence Intervals for three best algorithms tested in Experiment 1 as a function of ($\log_2$) number of surfaces in an image.

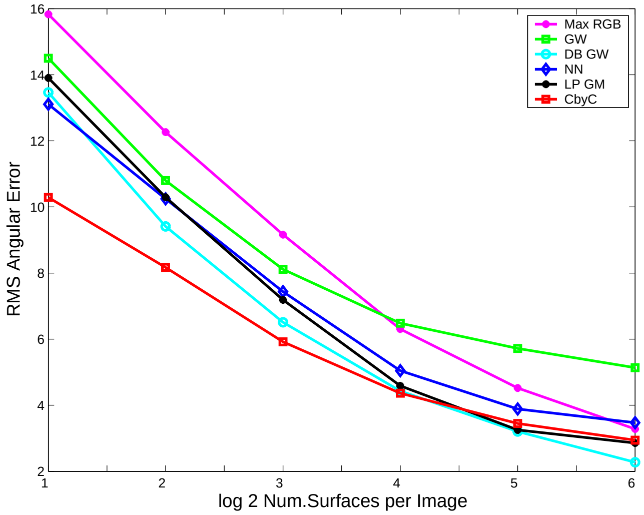
Fig. 5. Median angular error for each of the six algorithms tested in Experiment 2 as a function of ($\log_2$) number of surfaces in an image.

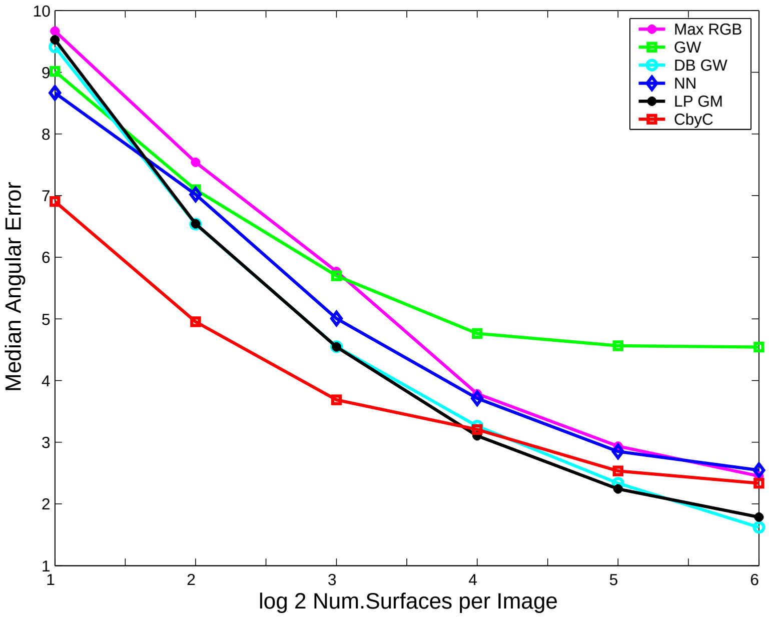

Number of Images
0
50
100
150
200
250
300
350
−0.2
−0.15
−0.1
−0.05
0
0.05
0.1
0.15
0.2
0.25
r−chromaticity error

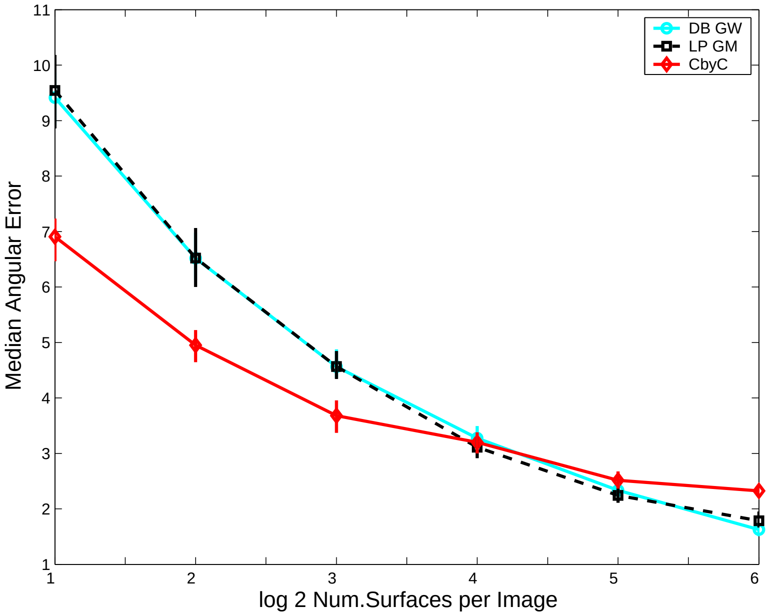

Median Angular Error
1
2
3
4
5
6
7
8
9
10
log 2 Num.Surfaces per Image
Max RGB
GW
DB GW
NN
LP GM
CbyC

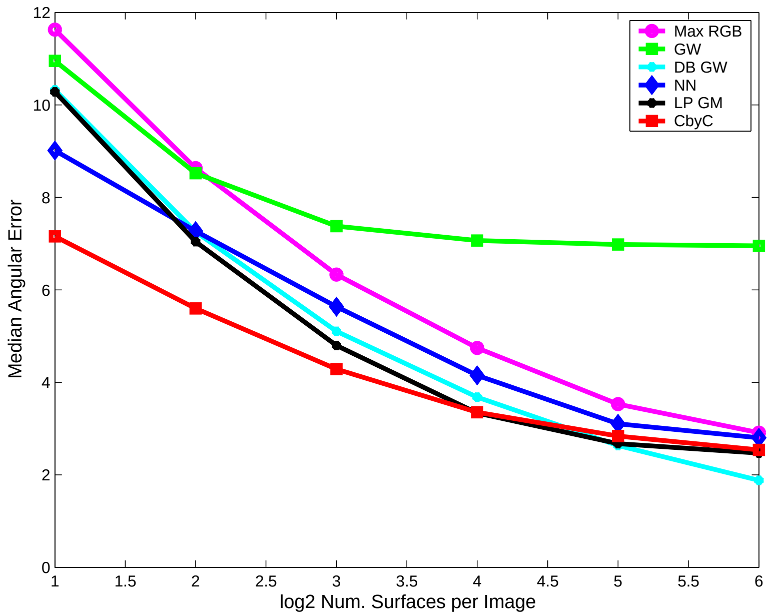

Number of Images
0
20
40
60
80
100
120
140
160
0
5
10
15
20
25
30
Angular Error


a
Quantiles of Angular Errors
−20
−10
0
10
20
30
40
50
−4
−3
−2
−1
0
1
2
3
4
Unit Normal Quantiles


RMS Angular Error
2
4
6
8
10
12
14
16
1
2
3
4
5
6
log 2 Num.Surfaces per Image
Max RGB
GW
DB GW
NN
LP GM
CbyC


Max RGB
GW
DB GW
NN
LP GM
CbyC
Median Angular Error
log 2 Num.Surfaces per Image
1
2
3
4
5
6
7
8
9
10


DB GW
LP GM
CbyC
Median Angular Error
log 2 Num.Surfaces per Image
1
2
3
4
5
6
7
8
9
10
11



Median Angular Error
log2 Num. Surfaces per Image
0
2
4
6
8
10
12
1
1.5
2
2.5
3
3.5
4
4.5
5
5.5
6
Max RGB
GW
DB GW
NN
LP GM
CbyC