Is Machine Colour Constancy Good Enough?

Brian Funt, Kobus Barnard and Lindsay Martin
School of Computing Science,
Simon Fraser University
Burnaby, British Columbia
Canada V5A 1S6
{funt, kobus, colour}@cs.sfu.ca

Abstract. This paper presents a negative result: current machine colour constancy algorithms are *not* good enough for colour-based object recognition. This result has surprised us since we have previously used the better of these algorithms successfully to correct the colour balance of images for display. Colour balancing has been the typical application of colour constancy, rarely has it been actually put to use in a computer vision system, so our goal was to show how well the various methods would do on an obvious machine colour vision task, namely, object recognition. Although all the colour constancy methods we tested proved insufficient for the task, we consider this an important finding in itself. In addition we present results showing the correlation between colour constancy performance and object recognition performance, and as one might expect, the better the colour constancy the better the recognition rate.

1 Introduction

We set out to show that machine colour constancy had matured to the point where it would be useful in other aspects of machine vision. Since all the different colour constancy algorithms require only a fraction of a second to run, they could be practical so long as their results were sufficiently accurate. 'Sufficiently accurate' begs the question—"Accurate enough for what?"—so we needed to choose a representative task that would provide an answer to the question and simultaneously give us a way to measure accuracy.

In the past the performance of colour constancy algorithms has been reported in terms of average angular error or RMS error between the predicted and target images [1]. In this paper, we test colour constancy by putting it to use in colour-based object recognition. The object recognition strategy is Swain and Ballard's "colour indexing" method [2], which is based on comparing histograms of the distribution of image colours. Colour indexing fails miserably when the ambient light illuminating the object to be recognized differs from that used in constructing the database of model images. Swain and Ballard suggest using colour constancy preprocessing as a way of addressing this problem; however, it has since been solved by introducing illumination-independent representations (e.g., relative colour instead of absolute colour [3] or by moment-based representations of colour histograms [4]). Nonetheless, if colour constancy methods work then it seems a natural task for them

to be used in preprocessing images prior to indexing as Swain and Ballard originally suggested.

Clearly, the fact that colour indexing is sensitive to variations in the ambient scene illumination is to be expected since its an entirely colour-based method and the scene illumination directly affects the image RGB colour[1]. The question we address here is whether or not existing colour constancy algorithms are effective enough at generating illumination-independent colour descriptors that colour indexing will work under the typical range of scene illuminations that are encountered in practice such as daylight, tungsten light, and fluorescent office lighting. Since our goal is to test colour constancy, not to develop a new and improved object-recognition scheme, we use colour indexing without modification.

The outline of the paper is as follows: first colour indexing and the importance of colour constancy for it will be discussed; then the method for colour correction given a good estimate of the illumination will be considered; this will be followed by a brief description of each of the colour constancy methods (greyworld, white-patch retinex, neural net, 2D gamut-constraint, and 3D gamut-constraint); next is the experimental setup and a description of the database of test images; following this are results and discussion.

2 Colour Indexing and Colour Constancy

The task for a machine *colour constancy* algorithm is to generate illumination-independent descriptors of the scene colours measured in terms of the camera RGB coordinates. The camera output is affected by the surface reflectance and the illumination. For the red channel we have

$$R(x,y) = \int E(\lambda)S(x,y,\lambda)C_{\mathrm{R}}(\lambda) \tag{1}$$

where $C_{\mathrm{R}}(\lambda)$ is the spectral sensitivity of the camera's red channel (similar equations for the green and blue channels $G(x,y)$and $B(x,y)$), $E(\lambda)$ is the spectrum of the incident illumination, and $S(x,y,\lambda)$ is the spectral reflectance of the surface.

We assume that the relative spectral power distribution of $E(\lambda)$ is spatially invariant (its intensity may vary), although some colour constancy methods have been developed that exploit spatial variation in illumination [5]. Surface colours as they would have appeared under some chosen 'canonical' illuminant will be used as illumination-independent colour descriptors. Hence the machine colour constancy problem can be expressed as that of deriving an image of the scene as it would appear under the canonical illuminant $\mathrm{RGB}_{\mathrm{canonical}}(x,y)$ given the image of the scene $\mathrm{RGB}_{\mathrm{unknown}}(x,y)$ under the unknown illuminant. The mapping has only to account for the change in relative spectral power distribution between the unknown and canonical illuminants.

---

[1] RGB space defined as the output of our SONY DXC-930 3-CCD colour video camera. Strictly speaking 'colour' is what a human observer perceives, but in this paper we will also use it to refer to a pixel's RGB.

Many colour constancy methods estimate only the chromaticity of the colours under the canonical illuminant and ignore the intensity component. There are many ways of normalizing the RGB to eliminate the effect of intensity of which we will use two different ones here. Colour indexing will be based on the standard chromaticity coordinate space:

$$r=R/(R+G+B); \quad g=G/(R+G+B) \tag{2}$$

For colour correction and the 2D gamut-constraint algorithm discussed below, we will use

$$r=R/B; \; g=G/B \tag{3}$$

Colour constancy algorithms will be used to convert between chromaticity 'images', in other words from the chromaticity under the unknown illumination $rg_{unknown}(x,y)$ they will provide an estimate of what the chromaticity $rg_{canonical}(x,y)$ would have been under the canonical illumination.

Colour indexing is performed using 2-dimensional chromaticity histograms. Swain and Ballard did the majority of their tests using RGB but they included some tests with rg-chromaticity space. The method is quite simple. First a database of model (chromaticity) histograms is created from images of the objects that we wish the system to recognize. The objects need to be separated from the background before the database is built. This segmentation can be done manually if need be. Given an image of an object to be recognized—call it the 'test' object—its chromaticity histogram is determined. Unlike the case for the model objects, the test object does not need to be separated from the image background. The test histogram *T* is then intersected with each model histogram *M* in the database, where intersection is defined as,

$$H(T,M) = \sum_{j=1} \min(T_j, M_j) / \sum_{j=1} M_j \tag{4}$$

The model with the highest histogram intersection score is used to identify the unknown object.

In our implementation the chromaticity histograms are 16x16. This sampling might be too coarse for a very large image database, but for our purposes the coarse sampling should help tolerate inaccuracies in colour constancy.

3 Colour Constancy and Colour Correction

We test 5 different colour constancy algorithms: greyworld, white-patch retinex, 2D gamut-constraint, 3D gamut-constraint and neural network. These algorithms all either estimate the colour of the incident illumination and then use that estimate to transform the image colours to canonical colour descriptors, or as in the case of the gamut-constraint algorithms, they estimate the transformation directly. We do not test the Maloney-Wandell [6] algorithm since previous tests [7] have shown it to perform very poorly, often worse than doing no colour constancy at all.

The colour correction step is in each case based on a diagonal model of illumination change. Other names for the diagonal model are von Kries adaptation and coefficient rule [8] . The diagonal model simply states that the effect of moving from one scene illuminant to another can be modeled by scaling the R, G, and B channels by independent scale factors. These scale factors can be written as the elements of a diagonal matrix. Previous work has shown that the diagonal model works almost as well as a full 3x3 linear model for typical scene illuminants [9]. In particular, for the type sensors found in our video camera, which have relatively narrow band and non-overlapping sensitivity functions, the diagonal model works very well.

Some of the algorithms work in a 2-dimensional chromaticity space. In this case a 2x2 diagonal transform can still model the change in chromaticity caused by moving between illuminants. For the diagonal model to hold, the two-dimensional chromaticity coordinates must be those defined by Equation 3. Finlayson [10] shows that this choice of chromaticity coordinates is crucial in preserving, in 2-dimensional coordinates, the diagonal model of illumination change that was present in the original 3-dimensional coordinates.

Each of the colour constancy algorithms we test will be described briefly in turn. The version of the greyworld algorithm we use compares the average of all the RGB in the image to a 50% ideal grey under the canonical, i.e., to RGB_{grey} given by $RGB_{grey}=(^1/_2)*RGB_{canonical}$. The diagonal scale factors for colour correction then are simply $R_{grey}/R_{average}$, $G_{grey}/G_{average}$ and $B_{grey}/B_{average}$.

The white-patch retinex algorithm compares the RGB of white under the canonical to the maximum found in each of the 3 image bands separately. There are many different variants of retinex and our white-patch version corresponds to the infinite-path-without-reset case described by Brainard and Wandell [11]. It differs from the retinex described by McCann et al. [12] Once the maximum in each colour channel is found, the diagonal scale factors for colour correction are simply $R_{canonical}/R_{max}$, $G_{canonical}/G_{max}$ and $B_{canonical}/B_{max}$.

Previously studies [1] have shown the various gamut-constraint methods [8, 10, 13] to be some of the best performing machine colour constancy methods. The gamut-constraint method derives constraints on the ambient illumination by evaluating the differences between the gamut of colours found in the image and those of a canonical gamut. For our experiments we constructed the canonical gamut from a database of hundreds of reflectance spectra from a wide variety of common objects. The canonical gamut is given by the convex hull of the set of RGB values that would have arisen if these reflectances were to be illuminated by the canonical illuminant. To understand the gamut-constraint method, consider an RGB triple **a** arising in an image of a scene under some unknown illumination. What does **a**'s presence reveal about the illumination? Since the canonical gamut represents the full set of RGB's ever expected to occur, the same spot under the canonical illuminant must correspond to some RGB inside the canonical gamut. However, since **a** has been obtained under an illuminant different from the canonical one, it may no longer lie within the canonical gamut. The set of diagonal transformations mapping **a** back to the canonical gamut represents the set of possible unknown illuminations.

Consider as a simple example a database having only 4 reflectances resulting in the 4 chromaticities (1,2), (2,5), (4,4), (4,3) under the canonical illuminant. The canonical gamut is defined by the convex hull of these 4 points, which in this example happen to all be on the hull. Intensity in this case is eliminated by moving to the two-dimensional chromaticity coordinates defined in Equation 3.

Consider an image RGB triple, **a**=(6,10,2), converted to chromaticity coordinates (3,5), which turns out not to lie within the canonical gamut. What does it take to map it to the canonical gamut? If we suppose that **a** corresponds to one of the 4 known reflectances, say that represented by (1,2) in the canonical gamut, then to map it there requires a scaling of the first component by 0.33 and the second component by 0.4. On the other hand, it might correspond to the canonical gamut point (2,5) in which case a scaling of 0.67 and 1.0 is needed. The other 2 canonical gamut points yield 2 more scaling pairs. Figure 1 plots the four mappings as filled diamonds.

Of course it might have been the case that **a** corresponds to one of the points inside the convex hull of the canonical gamut. However, only linear scalings are involved, so mapping to those interior gamut points would only result in scalings within the interior of the convex hull (solid lines) of the mappings in Figure 1. The convex hull of the set of mappings, therefore, represents the complete set of mappings that could take **a** into the canonical gamut. . Each point within the convex hull in Figure 1 represents a different hypothesis about the unknown illumination. Each point models the change in (r,g) created by moving from the canonical illumination to a possible unknown illumination.

The convex hull in Figure 1 therefore expresses the constraints that finding **a** in the image imposed on what the unknown illumination might be. The illumination must be represented by one of the points within the convex hull because these are all the illuminations that could possibly have resulted in one of the colours in the canonical gamut appearing as **a**.

One RGB **a** yields constraints on the unknown illumination and a second one **b** will yield further constraints. Suppose the second chromaticity is (2,4), then the mappings taking **b** to the hull vertices of the canonical gamut are as shown by the

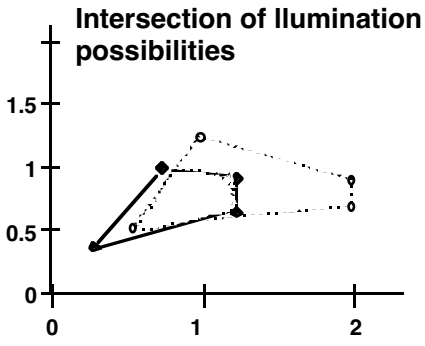

Figure 1. Solid lines show the convex hull of four diagonal mappings (0.33,0.4), (0.67,1.0), (1.33,0.8) and (1.33,0.6) that transform chromaticity (3,5) to the 4 vertices of the canonical gamut. Dashed lines show convex hull of mappings taking (2,4) to the canonical gamut. The mappings describing the unknown illumination are therefore restricted to the intersection (shaded) of the two convex sets

open circles in Figure 1 superimposed on the mappings for **a**. Since both **a** and **b** appear in the image, and by assumption, both scene points are lit by the same unknown illumination, the unknown illumination must be represented by one of the mappings in the intersection of the two convex hulls. All other candidate illuminations are eliminated from consideration.

The convex hull of the set of distinct rg's in the image is called the *image gamut*. Each vertex of the image gamut will yield some new constraints on the unknown illumination that can be intersected with the constraints obtained from the other vertices. Once the mapping constraint set has been established some heuristic method must be used to pick one of the remaining candidates as the estimate of the unknown illumination. We have used the hull centroid as the final estimate.

Forsyth's CRULE [8] gamut-constraint method does not consider the possibility of illumination constraints. The experimental results reported below are based on the gamut-constraint method with added illumination constraints [10]. Measurement of lots of different light sources reveals quite a restricted gamut. Our sampling of illuminants includes 100 measurements of illumination around the university campus, including both indoor and outdoor illumination. Some inter-reflected light was included such as that from concrete buildings and light filtering through trees, but illumination that was obviously unusual was excluded. The resulting illumination gamut is reformulated in terms of the set of diagonal transformations mapping each illuminant to the canonical illuminant. In this form it can be intersected with the constraints from the image gamut to further constrain the estimate of the unknown illumination.

Gamut-constraint methods can be carried out either in a 2D chromaticity space or a 3D RGB space. The 3D gamut-constraint method is just like the 2D case except that the constraint sets are now polyhedral convex hulls. Potentially the 3D case can be used to estimate the surface brightness in addition to surface chromaticity, but for our experiments we do not require the brightness information, so after running the 3D method we convert back to 2D chromaticity space. It should be noted that carrying out the gamut-constraint method in 3D and converting the result to 2D is not equivalent to carrying it out in 2D in the first place. In addition, when the 3D method is being used to estimate chromaticity, the final estimate is made by maximizing the volume of the intersection set. This method of choosing the final estimate originates in [8] and gives better chromaticity estimates than the hull centroid in the 3D case.

4 Neural Network Colour Constancy

Previously reported results [1] have shown good performance using a neural network for colour constancy. The network estimates the illuminant chromaticity based on the gamut of colours present in the image. The neural network is a Perceptron [14] with one hidden layer and an input layer consisting of 1000 to 2000 binary inputs representing the chromaticity of the RGB's present in the scene. The hidden layer has a much smaller size, usually about 16-32 neurons and the output layer is composed of only two neurons. Each image RGB from a scene is transformed into standard rg-chromaticity space (Equation 2) which then is coarsely, but uniformly, discretized and

presented to the network's input layer. The input is binary indicating either the presence or absence of the corresponding chromaticity in the image. The output layer of the neural network produces the values r and g (in the chromaticity space) of the illuminant. The output values are real numbers ranging from 0 to 1.

We trained the network using a back-propagation algorithm without momentum[15] on thousands of synthetic images generated by randomly selecting 1 illuminant and from 1 to 60 reflectances from our database of surface reflectances and illuminants, and then integrating them with the with the spectral sensitivity functions of our camera in accordance with Equation 1. During the learning phase the network is provided the image data along with the chromaticity of its illuminant.

5 The Test Images

The images used for our experiments are of 11 different, relatively colourful objects. (Figure 2 shows the objects). The pictures were taken with a Sony DXC-930 3-CCD colour video camera balanced for 3200K lighting with the gamma correction turned off so that its response is essentially a linear function of luminance. The RGB response of the camera was calibrated against a Photoresearch 650 spectraradiometer. The aperture was set so that no pixels were clipped in any of the three bands (i.e. R,G,B<255). Since most of the images had some specular highlights, reducing the aperture in this manner left much of each image quite dark. To overcome this problem we also extended the dynamic range of the images roughly 5-fold by averaging 25 frames (and storing the result as floating point images). Recording the images in this way produced a more versatile image database since it can then be used to simulate both the effect of brightening the images by increasing the aperture and the effect of clipping bright spots.

We took images under 5 different illuminants using the top section (the part where the lights are mounted) of a Macbeth Judge II light booth. The illuminants were the Macbeth Judge II illuminant A, a Sylvania Cool White Fluorescent, a Philips Ultralume Fluorescent, the Macbeth Judge II 5000 Fluorescent, and the Macbeth Judge II 5000 Fluorescent together with a Roscolux 3202 full blue filter, which produced a illuminant similar in colour temperature to a very deep blue sky. The effect created by changing between these illuminants can be seen in Figure 3 where the same ball is seen under each of the illuminants. The illuminant spectra are plotted in Figure 4.

Two sets of images were taken. For the "model" set, we took images of each object under each of the five illuminants, without moving the object. In other words, we have 11 groups of 5 registered images. The "test" set is similar, except that the object was purposefully moved before taking each image. With these two sets of images we are then able both to evaluate colour indexing under different scene illuminants with and without changes in object position. In total, 110 images were used.

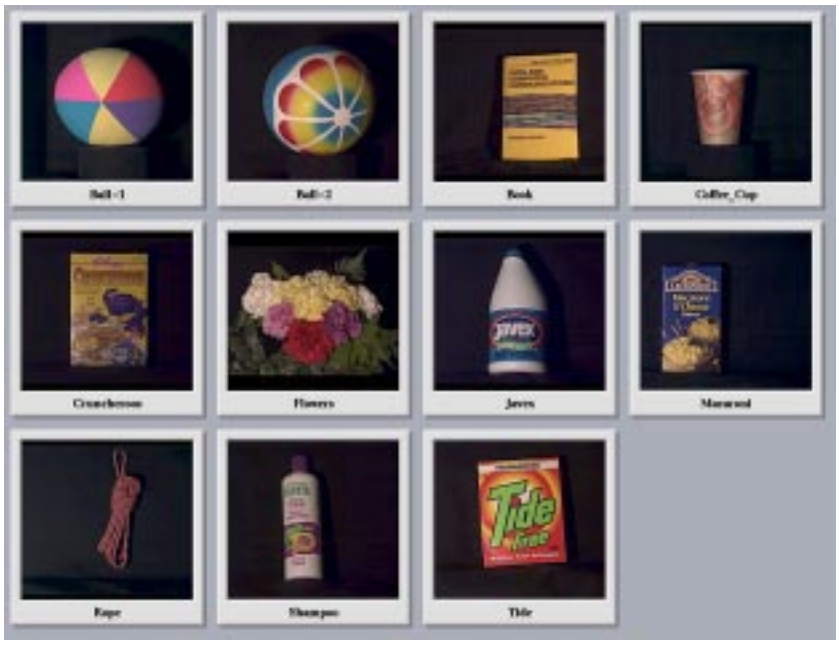

Figure 2. The 11 objects in the image database as seen under a single illuminant

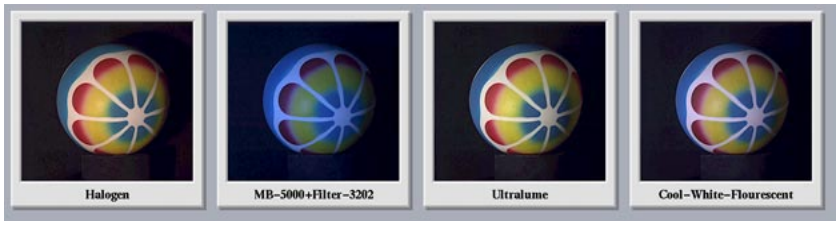

Figure 3. Ball-2 as seen under 4 of the 5 illuminants

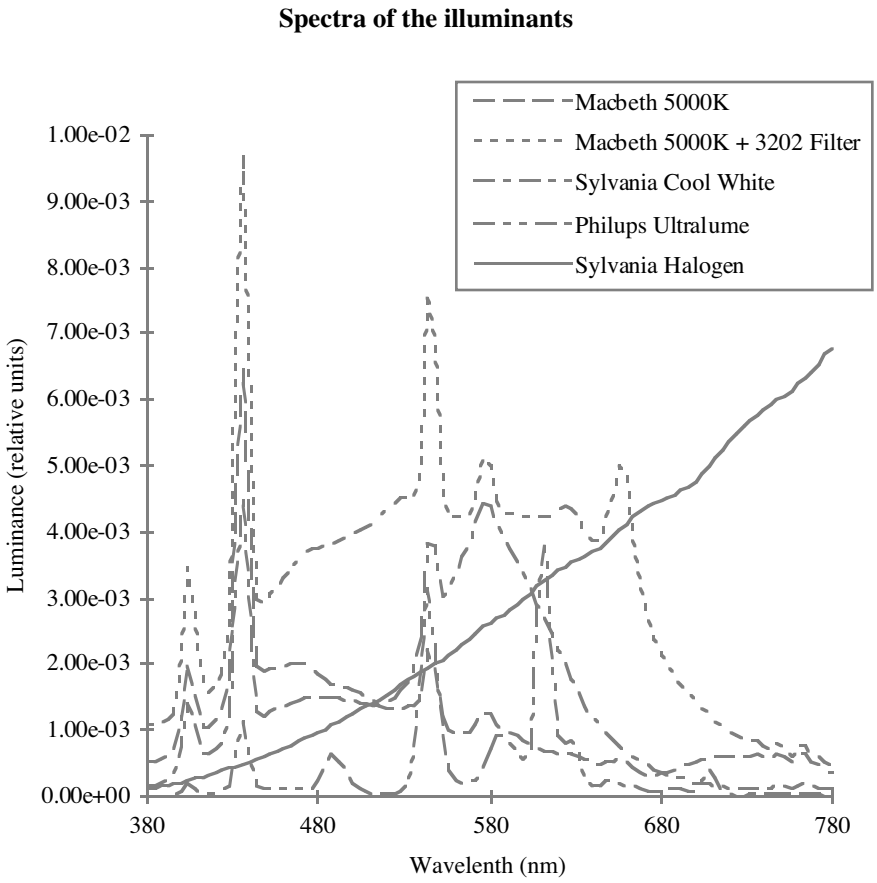

Figure 4. Spectra of the five test illuminants as measured by a Photoresearch spectrometer

Before using the images, they were first adjusted according to our camera calibration model. To do this we first subtract the variation of the background from the average background level, as determined by averaging a large number of images with the lens cap on. Then we adjusted pixels darker than a certain amount with a pre-established look-up table to make up for a small non-linearity in the camera. Finally, we subtracted the per-channel intercept of the camera linearity data from the images. The result is an image which is closer to one taken by an "ideal" camera, under the model that the RGB values are simply integrals as in Equation 1 of the incoming spectra multiplied by sensor sensitivity functions. When computing image histograms, the data is further cleaned up by averaging 5-by-5 blocks of pixels and excluding very dark pixels.

7 Results

Our experiments confirm the obvious hypothesis that colour constancy is likely to improve colour indexing in situations where the illumination impinging on the test object is different from that used in constructing the model database. Figure 5 shows a clear correlation between colour indexing performance and colour constancy error.

The recognition performance measure used in Figure 5 is based on a weighted average of colour indexing's rankings. During the recognition phase, colour indexing calculates match strengths for each model in the database. If the strongest match is in fact the correct object, then we say that we have a rank one match. If the correct object

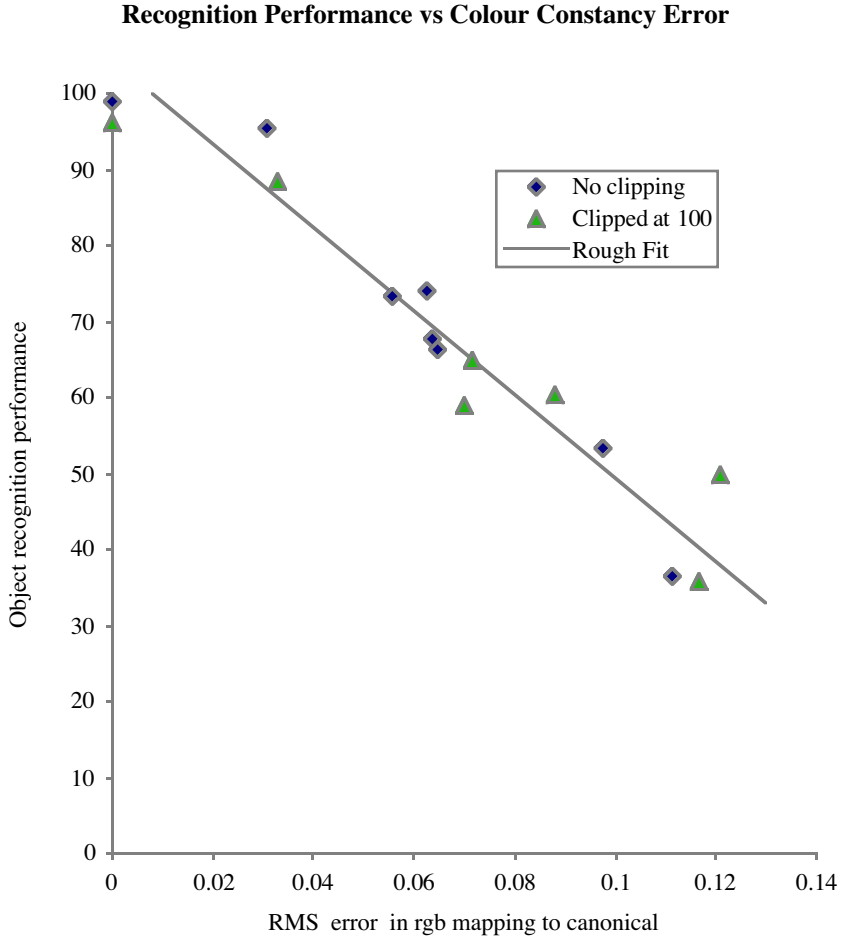

Figure 5. Object recognition performance as a function of colour constancy error.

is the algorithm's second choice, then we have a rank two match, and so on. For each algorithm, we obtain a percentage of the total (220 possible) matches by rank. To distill these results into a single representative value, we use a weighted sum of the percentages of the first three ranks: the weight for rank one is one, the weight for rank two is 1/2, and the weight for rank three is 1/3. Matches beyond rank 3 are considered failures and count as zero.

The measure for colour constancy error is based on a comparison of corrected and "target" images. Using the set of registered model images, we compute the root mean square difference on a pixel-by-pixel basis taken across the entire image in chromaticity between the target image—the one taken under the canonical illuminant—and the colour corrected image. Of the 5 illuminants, 4 are adequate for use as a canonical illuminant. One illuminant creates such a blue cast in the images that many of the red and green intensities are very low and possibly less reliable. As result for each colour constancy algorithm we obtain 4x55=220 colour constancy results, which are averaged to produce the data plotted as filled diamonds in Figure 5.

As noted above, our data set has an extended dynamic range with the images being purposefully underexposed in order to prevent any clipping. Visual checking by an impartial colleague suggested that exposure levels for normal viewing would be on average 2.5 times higher. Thus we simulated image capture at this level by scaling by 2.5 and thresholding anything greater than to 255 to 255. We then re-ran the matching experiments; the results are plotted with filled triangles in Figure 5.

Part of our motivation for experimenting with the clipping level was the unexpectedly good performance of the white-patch retinex method which was comparable to the gamut-constraint and neural net methods This was surprising given comparative results reported by Funt et al. [1]. Upon reflection, however, one might expect retinex to do relatively well given the special unclipped nature of our images. Clipping often is the cause of Retinex's failure because it relies on the brightest pixels being accurate, and clipped pixels clearly are not accurate. Specular highlights provide excellent clues to retinex as to the colour of the illuminant, again, providing they are not clipped. Our images generally do contain unclipped specularities. Even without specularities, preserving the maxima in each channel definitely will increase the performance of white-patch Retinex.

Under the more usual case, where either the human user or automatic aperture control has adjusted the capture process to obtain a pleasing image, there are invariably clipped pixels. In this case retinex starts to break down because by definition, the maximum value in at least one channel is a bad data point. To verify that this degradation does occur, we simulated clipping at various levels. The result is plotted in Figure 6. Here it is clear that as the clipping level increases, the retinex algorithm degrades much more quickly than the others, and when clipping is at the level of 75-100—consistent with the scaling of 2.5 used above—then its performance is close to previously reported results. The performance of all the algorithms in the non-clipped case is shown in Table 1.

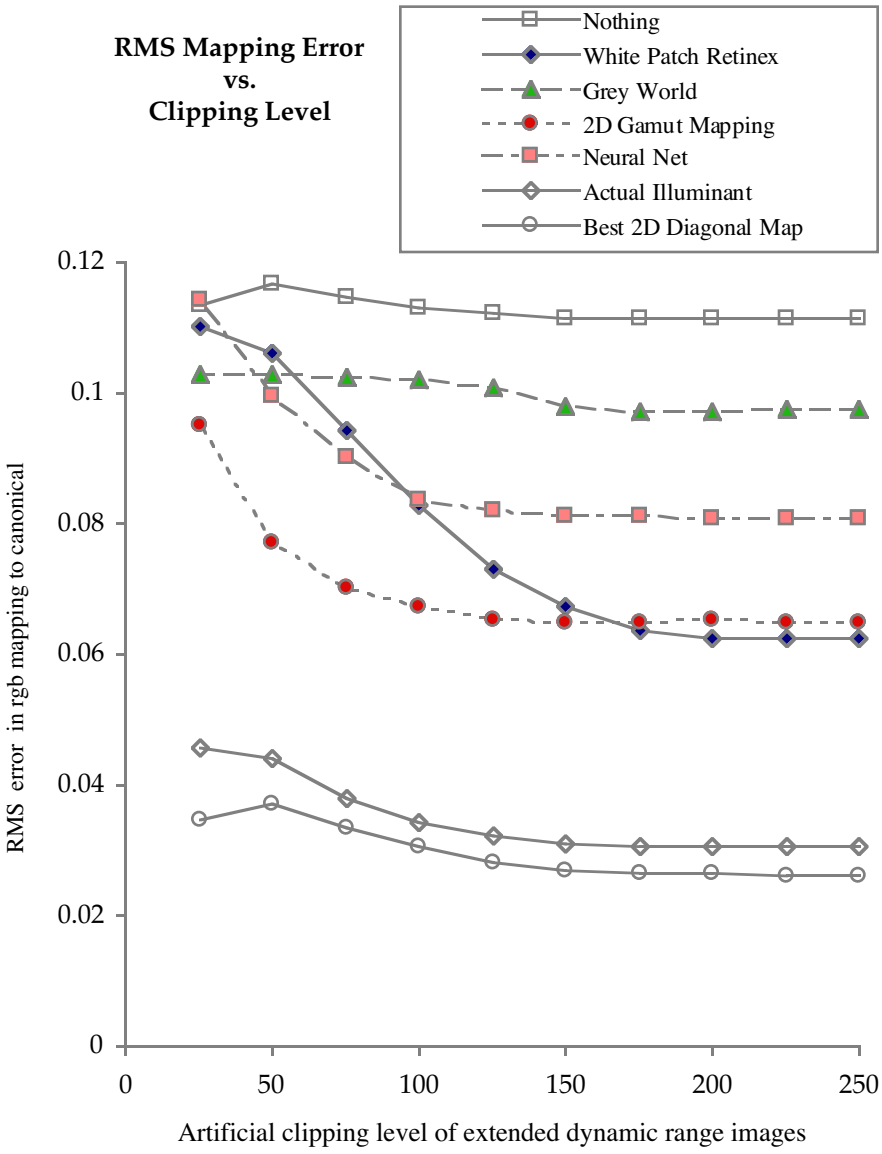

Figure 6. The RMS mapping error versus the artificially imposed clipping level. The RMS mapping error is the RMS difference between the chromaticity of the corrected images and the target image. The curve for the "algorithms" which do not explicitly depend on the data, such as "nothing" are almost flat, but not exactly flat, because as the clipping increases, we take the RMS difference between fewer and noisier data points. As conjectured, the white-patch Retinex method degrades much faster than the other algorithms.

Our original question was: Is machine colour constancy good enough? Based on our results, we feel that the answer is no. Table 2 shows the indexing performance of colour indexing using each algorithm for preprocessing. The best performer finds only 67% of the objects (rank 1 matches); whereas, the results (92% rank 1) based on using the actual illuminant RGB for colour correction indicate that both colour indexing and the diagonal model of illumination change will together support much better performance. Thus we conclude that we still have some distance still to go before machine colour constancy is up to the task of supporting object recognition.

In terms of our methodology, we feel that we gave machine colour constancy every reasonable chance, and thus the results should be considered closer to the "best case" than the "worst case". For example, we used illumination with spatially

Algorithm	RMS Error
Nothing	0.1114
Actual	0.0306
White-Patch Retinex	0.0625
Greyworld	0.0975
2D Gamut-Constraint	0.0649
3D Gamut-Constraint	0.0555
Neural Net	0.0634

Table 1: Colour Constancy Error by Algorithm: The error is the RMS difference in chromaticity between mapped image and registered target image. The results are the average of 220 results, being the estimated maps for 55 images to each of 4 illuminants. "Nothing" denotes doing no colour constancy and simply using the input image as the output image. "Actual" denotes using the true unknown illuminant RGB to do colour correction based on the diagonal model.

Rank	1	2	3	4	5	6	7	8	9	10	11
Nothing	28.4	10.2	9.1	9.1	8.5	6.8	4.5	6.2	5.7	4.5	6.8
Perfect	97.7	2.3									
Actual	92.3	5.5	1.4	0.9							
Retinex	67.7	8.6	5.9	1.8	2.7	1.8	1.4	3.2	5.5	0.9	0.5
Grey world	46.4	9.1	7.3	8.2	2.3	5.0	1.8	2.3	5.9	9.5	2.3
2D Gamut	60.0	10.9	8.6	5.5	1.8	3.6	2.3	2.3	1.8	1.4	1.8
3D Gamut	67.3	8.6	5.5	2.3	2.7	3.2	3.2	1.8	2.3	0.5	2.7
Neural net	61.4	7.7	7.7	4.1	1.8	2.3	1.4	1.4	1.8	5.9	4.5

Table 2: Number of matches by rank in percent. The total number of attempted matches is 220 for each except for "perfect" where only 44 matches make sense. "Perfect" indicates the case where the test illumination is the same as the canonical illumination and hence there is exact "colour constancy."

uniform chromaticity and were careful to remove noise through temporal and spatial averaging. We have also taken some trouble to develop a good camera model as required by some of the algorithms. Finally, the database was relatively small, and we avoided bad matches due to colours appearing coincidentally in the background by placing the objects on black cloth.

Having said that, we wish to emphasize some aspects of the experiment that were not open to compromise. First, and foremost, the data is real image data, and the objects are random everyday objects as opposed to, for example, planer non-specular "Mondrians". In particular, the object database includes several objects with a significant fluorescent component (e.g., the Tide box image). The fluorescent component does not change in the same way as the matte component which presents a problem for the diagonal model. It is perhaps also of some significance that the images were taken by a research assistant who had little understanding of the intended purpose of the experiment—possibly eliminating any unintended bias in the choice of objects to test. Finally, the illuminants represent quite a dramatic range in terms of what is usually encountered in common natural lighting and standard man-made lighting situations without adding the unusual effects of things like heavily filtered theater lights.

8 Conclusion

We tested machine colour constancy algorithms using the computer vision task of colour-based object recognition based on colour histogram intersection. The colour constancy algorithms did not perform as well as expected based on previous results with colour balancing images. We expected that colour constancy processing would provide colour descriptors that would be accurate enough that colour indexing performance would be close to that obtained when there is no change in the ambient illumination. As shown in Table 2, this is did not turn out to be the case. Colour constancy pre-processing did, however, yield a significant improvement over doing no pre-processing, it simply was not enough of an improvement. Figure 5 shows that the degree of improvement in histogram matching appears almost linearly related to the RMS error in colour prediction.

The results of Brainard et al. [16] indicate that human colour constancy is not all that accurate and state (p. 2101) "Our results represent neither complete constancy nor a complete absence of constancy." The results of our experiments raise the question as to whether or not human colour constancy would be sufficiently accurate for histogram-based object recognition?

We take the current state of machine colour constancy performance as a challenge for future research. It is clear that without RMS errors in colour prediction under 0.04, a typical vision task such as object recognition can not be based on absolute descriptors (as opposed to relative ones, like ratios) of colour.

References

1. Funt, B.V. and Cardei, V., Barnard, K., Learning Color Constancy, *IS&T Fourth Color Imaging Conference*, Scottsdale, Nov. 1996.
2 Swain, M., and Ballard, D., "Color Indexing," *Int. J. Comp Vision*, 7:11-32, 1991.
3. Funt, B.V., and Finlayson G.D., Color Constant Color Indexing, *IEEE Trans. Patt. Anal. and Mach. Intell,* 17(5), May 1995.
4. Healey, G. and Slater, D. "Global Color Constancy: recognition of objects by use of illumination invariant properties of color distributions," *J. Opt. Soc. Am. A,* 11(11):3003-3010, Nov. 1994.
5. Barnard. K., Finlayson, G., Funt, B., Colour Constancy for Scenes with Spectrally Varying Illumination, *ECCV'96 Fourth European Conference on Computer Vision*, Vol. II, pages 3-15, April 1996.
6. Maloney, L. and Wandell, B., "Color constancy: a method for recovering surface spectral reflectance," *J. Opt. Soc. Am. A,* 3:29-33, 1986.
7. Finlayson, G., Funt, B. and Barnard, J., Colour Constancy Under a Varying Illumination, *Proc. Fifth Intl. Conf. on Comp. Vis.,* Jun 1995.
8. Forsyth, D., A novel algorithm for color constancy, *Intl. Journal of Computer Vision* 5:5-36, 1990.
9. Finlayson, G., Drew, M., and Funt., B., Color Constancy: Generalized Diagonal Transforms Suffice, *J. Opt. Soc. Am. A,* 11(11):3011-3020,1994
10. Finlayson, G. Color in Perspective, PAMI Oct. 1996. Vol. 18 number 10, p1034-1038
11. Wandell, B.A. Analysis of the Retinex Theory of Color Vision, *Journal of the Optical Society of America A,* V 3, pp.1651-1661, Oct. 1986.
12. McCann, J.J., McKee, P., and A Taylor, T.H., Quantitative Studies in Retinex Theory, *Vision Res. Vol.* 16 pp. 445-458, 1976.
13. Kobus Barnard, "Computational colour constancy: taking theory into practice," MSc thesis, Simon Fraser University, School of Computing (1995).
14. Hertz, J., Krogh, A., and Palmer, R.G. *Introduction to the Theory of Neural Computation*, Addison-Wesley Publishing Company, 1991.
15. Rumelhart, D.E., Hinton, G.E., and. Williams, R.J, Learning Internal Representations by Error Propagation in D.E. Rumelhart & J.L. McClelland (Eds.), *Parallel Distributed Processing: Explorations in the Microstructure of Cognition.* Vol.1: Foundations. MIT
16 Brainard, D., Brunt, W., and Speigle, J., Color Constancy in the nearly natural image. I. Asymmetric matches, *Journal of the Optical Society of America A,* V 14, pp.2091-2110, Sept. 1997.