

# Computational Uses of Colour

Brian Funt and Vlad Cardei

## Abstract

Why do we have colour? What use is it to us? Some of the obvious answers are that we see colour so that we can recognise objects, spot objects more quickly, tell when fruit is ripe or rotten. These reasons make a lot of sense, but are there others? In this paper, we explore the things that colour makes easier for computational vision systems. In particular, we examine the role of colour in understanding specularities, processing interreflections, identifying metals from plastics and wet surfaces from dry ones, choosing foveation points, disambiguating stereo matches, discriminating textures and identifying objects. Of course, what is easier for a computational vision system is not necessarily the same for the human visual system but it can perhaps help us create some hypotheses about the role of colour in human perception. We also consider the role of colour constancy in terms of whether or not it is required for colour to be useful to a computer vision system.

## Introduction

Often in computer vision, colour has been treated as an unnecessary complication of an already complex problem. To process colour requires more data storage and on the face of it more computation; however, one of our premises in this paper is that the extra information provided by colour may simplify the processing rather than complicate it. To see the extent to which this might be true, we examine how colour has been applied in solving a variety of computer vision problems. Much of computer vision can be viewed as an effort to extract automatically information about a scene from a digitized image of it. What are the ways in which colour can help in this process?

In general, colour is a perception that people have and as such is quite difficult to quantify. In this paper, however, we will be talking about colour in the context of computer vision and will use 'colour' to refer to the more restricted notion of simply the digitized 3-band signal ( $R$ ,  $G$  and  $B$ ) obtained from a typical colour video camera. Each pixel location in the camera's image provides 3 measurements of portions of the spectrum of the light impinging on the camera's sensor at that point.

We can describe the camera's response to a given spectrum of light in terms of three sensitivity functions similar to the way in which there are 3 cone types in the human visual system each with its own sensitivity as a function of wavelength. In the case of the camera, however, the sensitivity functions are not necessarily the same as those of the human cones, and in fact, in general they differ significantly from the cones. Furthermore, different colour video cameras models tend to generate different  $RGB$  signals in response to the same input signal, so even in the case of computer vision the definition of colour is unfortunately underdetermined.

The light entering a camera depends on both the surface from which the light is reflected and the light which illuminates that surface. If the illuminating light changes, so will the light entering the camera and as a result so will the  $RGB$  values in the colour image the camera records. The same is of course true for light entering the eye; however, as experiments by Brainard (Brainard 1997) have shown, human subjects can make adjustments for changes in the incident illumination quite well. This ability to account for changes in illumination and see colours as approximately the same under different lighting conditions is generally known as colour constancy. It would seem helpful for a computer vision system also to have colour constancy, and we will discuss an approach to computational colour constancy below, but it turns out that many of the computational uses for colour do not in fact require colour constancy for their operation.

An important tool in computational colour analysis is the colour histogram. A colour histogram of an image provides a count, for each colour ( $RGB$ ), of the number of pixels in the image which have that colour. Since image pixels are all the same size, the histogram also describes, for each colour, the total image area covered with that colour. Often only binary

valued (i.e. 0 or 1) histograms are required, in which case the histogram indicates only whether or not a particular colour is present in the scene, not the amount of the colour present. Histograms discard all spatial information about the image--- we cannot tell from an image's histogram where the colours were located in the image, but rather only that they were there. Although often the *RGB* values are histogrammed directly, histograms are also commonly made using the values obtained in other colour spaces, such as a chromaticity space, derived from the original. It must be noted that, since the sensors are camera dependent, all resulting *RGB* values will also be camera dependent, so the same scene viewed with two different cameras will most probably generate two different images (i.e. *RGB* sets) and similarly two different colour histograms.

As the illumination changes, the colours in the image will also change. and so will the histogram of the image. This colour shift poses a problem concerning the stability of the colours being used. For example, the colours change as the illumination changes, then how can colour be used for object recognition? Clearly, if the colours change, the colour histograms will change as well. How can computational vision systems compensate for the change? One possibility is to imitate human colour constancy in some way and thereby provide an approximately illumination-independent description of colours; however, as we will show it is not always necessary to compute colour constant descriptors in order to discount the effects the of changes in illumination.

### **Colour-Based Object Recognition**

Swain and Ballard (Swain and Ballard 1991) showed that colour can be very useful for object recognition. They designed a technique called 'colour indexing' that compares the colour histogram of a given object to a list of histograms of objects previously stored in a database. This method is based solely on how well the histogram of the test object matches the histograms in the database.

The method works surprisingly well even though it completely ignores all spatial information (i.e. regarding object shape or geometry). As a consequence, the algorithm is not affected by any rotation, translation or scaling of the test object. However, even slight changes in

the scene's illumination—in intensity or chromaticity— adversely affect the algorithm's performance as can be seen in Figure 1. When the illumination changes, all colours in the image change, so the new colour histogram of the object no longer matches its histogram stored in the database. To overcome this problem, Swain proposed adding colour constancy as a pre-processing step to compensate for changes in the illuminant.

Instead of pre-processing for colour constancy, Funt and Finlayson (Funt and Finlayson 1995) proposed the idea of using relative colour instead of absolute colour as a way of circumventing the colour constancy problem. Their 'colour constant colour indexing' method cancels out the effects of the illumination by histogramming the ratios of colours at neighbouring pixels rather than the colours themselves. So long as the illumination varies smoothly, if at all, throughout the image then the ratios will be approximately independent of the illumination no matter what it is. This assumption is reasonable since in practice abrupt changes in the incident illumination rarely occur.

How does comparing neighbouring pixels eliminate the illumination effects? For many camera sensors the changing from one illumination to another can be approximated as scaling each of the  $R, G$  and  $B$  by different factors. For example, moving from indoor tungsten bulb lighting to the much bluer daylight will generally mean an increase of  $\alpha$ , say, in the  $B$  image band to  $\alpha B$ . For two neighbouring pixels, of the same or different colour, and considering just the  $B$  band for simplicity, if they have  $B$  components  $B_1$  and  $B_2$  under daylight, they will have components  $\alpha B_1$  and  $\alpha B_2$  indoors. However, the ratio of the two pixels will remain constant:

$$\frac{B_1}{B_2} = \frac{\alpha B_1}{\alpha B_2}$$

Modelling illumination's effects by a scaling such as  $\alpha$ , however, holds only approximately as can be seen from the fact that  $E(\lambda)$  can not be correctly divided out inside the integral

$$\frac{\int E(\lambda)S_1(\lambda)R(\lambda)d\lambda}{\int E(\lambda)S_2(\lambda)R(\lambda)d\lambda} \cong \frac{\int S_1(\lambda)R(\lambda)d\lambda}{\int S_2(\lambda)R(\lambda)d\lambda} \quad (1)$$

where  $S_1$  and  $S_2$  are the surface reflectances corresponding to the two neighbouring pixels,  $R_B$  is the sensor response of the  $B$  channel and  $E$  is the illumination. This approximation holds quite

well for ‘sharp’ sensors (Finlayson, Drew and Funt 1994b), which are what are found in many cameras.

The colour indexing and colour constant colour indexing methods show that colour can be very useful for object recognition in a computer vision system. Both methods result in very high, and very comparable, recognition rates: so while colour information is an important object recognition cue, it is not necessary to have the absolute, or true colour, but rather only a measure of the relative colours of neighbouring locations.

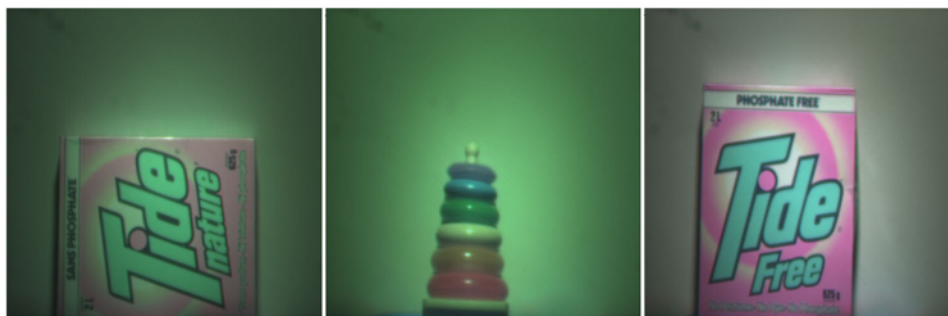


Figure 1. Left: object to be recognised. Centre: object found by simple colour indexing when illumination effects are not take into account. Right: object found by colour constant colour indexing method based on the illumination independence of colour ratios.

### **Interpreting Specularities**

Shiny, glossy surfaces are very common so it is important for a computer vision system to be able to recognise glossy regions and interpret them correctly; otherwise, algorithms such as those that extract shape from shading will be misled. The shiny, mirror-like component of surface reflection is known as the specular component and the bright highlights are called specularities. Shafer (Shafer 1985) pointed out that the specular component creates colour effects that can be used in understanding images of glossy surfaces.

A lot of work in computer vision has made the assumption that the surfaces seen in an image are primarily matte and can be modelled by the Lambertian reflectance model which states that surfaces appear equally bright from all viewing angles. While this is a useful

assumption from a theoretical standpoint, truly matte surfaces are rare and many not-very-shiny surfaces like a sheet of paper reflect a significant specular component. Unlike the matte component, the amount of specularity seen on the surface depends on the angle from which it is viewed and the relative position of the illuminating light.

What is interesting from the standpoint of colour analysis of specular reflection is that the specular and matte components generally will differ in colour. This is captured in Shafer's dichromatic model of reflection (Shafer 1985). Materials like plastics, but not metals, are inhomogeneous in that they are composed of a medium and a colorant which scatters the light. In such materials, incident light first interacts with the interface between the air and the material and some of the light is reflected by the interface without further penetrating the material. This part of the light reflection is called *interface reflection*. Interface reflection is often seen as a highlight or specularity and so it is also called *specular reflection*. The relative amount of reflected light at the interface as well as the reflection angle are predicted by Fresnel's laws. Since the index of refraction is relatively constant over the visible spectrum, the interface reflection is assumed to be constant with respect to the wavelength and consequently to have the same spectral composition as the incident light. Hence, the part of the light reflected by the interface will have the same colour as the illuminating light.

The remaining part of the incident light, which was not reflected at the interface, penetrates into the medium, is scattered by the colorant particles, and is eventually either absorbed, transmitted (if the material is not opaque) or re-emitted back through the interface. The portion of the light re-emitted from the body of the surface is called the *body reflection* component. Generally the body reflection component will differ in colour from the colour of the illuminating light because of the selective absorption of the colorant.

The dichromatic model of reflection states that the total radiance of reflected light is equal to the sum of two independent components: the radiance of the interface reflection and radiance of the body reflection. The relative amounts of the interface and body reflection components will vary with the angle at which the incident illumination hits the surface and the direction from which the surface is viewed. If the colours of the interface and body components are given by  $R_iG_iB_i$  and  $R_bG_bB_b$  respectively, then the colour seen by the camera will be:

$$RGB = m_i R_iG_iB_i + m_b R_bG_bB_b, \quad (2)$$

where  $m_i$  and  $m_b$  are the respective magnitudes of the two components.

Suppose we now consider a R-versus-G-versus-B plot of the colours found in an image containing only a single shiny surface. So long as the surface is not completely flat then as the surface curves, the position of each small patch of surface will vary with respect to the light and camera so that the relative magnitudes of the interface and body components will change. The dichromatic model predicts that all the colours will fall on a single plane in the R-versus-G-versus-B plot and they in fact all be contained within a parallelogram defined by the  $R_iG_iB_i$  and  $R_bG_bB_b$  colours. The magnitude coefficients  $m_i$  and  $m_b$  determine the position inside the parallelogram..

Specular reflection results in well defined distributions of image colours, so colour can help in the analysis of images of shiny surfaces by applying the dichromatic model. By fitting a plane and then a parallelogram to the distribution of plotted  $RGB$ 's, the colours of  $R_iG_iB_i$  and  $R_bG_bB_b$  of the interface and body components can be extracted. They are useful since the interface component, for example, tells us about the colour of the incident illumination. The components of the original image can also be separated to generate an image of the object as if it were completely matte and another as if it were completely mirror-like, both of which are useful in determining the shape of the object.

### **Interpreting Interreflections**

Light interreflected between surfaces often accounts for a significant amount of the illumination impinging on them. The amount can easily be as high as 15% and yet it seems that people rarely notice interreflections even though they can introduce significant colour shifts. Figure 2 shows quite an extreme example of interreflection where significant components of red and green occur on the white cup; nonetheless, the cup still appears basically white.



Figure 2. Example of interreflection.

In terms of a computer vision system, colour can be used as a clue in analysing images containing interreflections. Correctly interpreting interreflections can be very important since, for example, many computer vision models such as those that extract shape from shading information presume that all the illumination a surface receives comes directly from the light source.

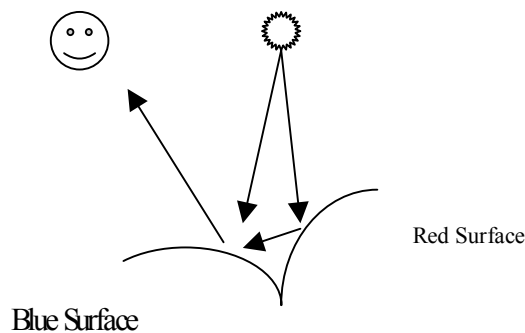


Figure 3. Light arriving at the eye or camera will consist of blue light reflected directly from the blue surface along with red-blue light reflected by both the red and blue surfaces.

Consider a scene that contains two non-coplanar surfaces of different colour, red and blue for example, but illuminated by a common light source as shown in Figure 3. Some of the red



light reflected from the red surface may impinge on the blue surface, which will then be lit by both the main light source and the red surface. Similarly, the blue surface may add some blue illumination to the red surface. In both cases the extra light interreflected between the surfaces will change the colour of those surfaces as seen by the camera. Ignoring the change in colour created by interreflection could easily adversely affect colour segmentation and object recognition algorithms. The interreflection process, of course, does not necessarily end after one iteration. Red light that hit the blue surface could again be interreflected back to the red surface, but the magnitude of these multiply reflected components drops quickly.

Analysis of the colour shifts created by interreflection turns out to have a lot in common with that for specular reflection. As with the case of specular reflection, the goal in interpreting interreflections is to separate the colour image into components describing those arising from the direct illumination of the light source and those arising from indirect illumination via interreflection.

While the most general model of interreflection would allow for an infinite number reflections between the two neighbouring surfaces, in practice a simplified one-bounce model works well in many situations and is much easier to analyse. The one-bounce model, based on a single interreflection of the light from one surface to the next, predicts that the light reflected from a surface consists of a linear combination of a no-bounce contribution and a one-bounce contribution. The no-bounce contribution is the one resulting from the illuminant being directly reflected from the surface, while the one-bounce contribution comes from the reflection of another surface.

The spectrum of the light  $C_A(x, \lambda)$  from a pixel at location  $x$  on surface  $A$  with reflectance  $S_A(\lambda)$  receiving light reflected from the light source of spectrum  $E(\lambda)$  and from surface  $B$  with reflectance  $S_B(\lambda)$  is described by the one-bounce model as:

$$C_A(x, \lambda) = \alpha_A(x)E(\lambda)S_A(\lambda) + \beta^{BA}(x)S_A(\lambda)E(\lambda)S_B(\lambda), \quad (3)$$

where  $\alpha_A(x)$  and  $\beta^{BA}(x)$  are the relative magnitudes of the two contributions. In terms of camera  $RGB$  rather than full spectra, this means that the final colour registered by the camera results from a no-bounce colour  $R_0G_0B_0$  and a one-bounce interreflection colour  $R_1G_1B_1$

$$RGB = \alpha_A R_0G_0B_0 + \beta^{BA} R_1G_1B_1 \quad (4)$$

This equation exactly parallels Equation (2) above so, as in the case of specularities, the colours created by interreflection will again lie within a parallelogram in  $RGB$  space.

The case of light reflected from  $A$  to  $B$  is symmetrical to that of light reflected from  $B$  to  $A$ . This is reflected in the symmetry of the second term in equation (3), so the one-bounce colour reflected from surface  $A$  to  $B$  matches that from  $B$  to  $A$ . The  $RGB$ 's from  $A$  and  $B$  will lie in a two separate parallelograms in colour space; however, the equality of the one-bounce colour between them implies that the two parallelograms will share a common edge. Hence, the one-bounce colour can be determined (Funt and Drew 1993) as the intersection of the two parallelograms. This can then be used to extract the no-bounce components and produce images representing the scene with or without interreflection effects. Hence as in the case of specularities, colour provides important information that helps in understanding interreflections.

### **Colour and Shape**

In computer vision there have been many different approaches to extracting the shape of surfaces from the shading of the surfaces. Most of these have assumed that the surfaces were of a single colour, but of course this assumption is generally violated in any practical scene. Using a grayscale, “black and white” image of a scene does not solve the problem for shape from shading algorithms, since of course the world is still coloured even if the image is not.

One use of colour in the context of shape from shading is in separating the changes arising in an image created by changes in shading (i.e. the amount of incident illumination) from those due to variations in surface colour. (Funt, Drew and Brockington 1992) propose an algorithm that recovers the intrinsic shading field from a coloured image. The shading field describes the changes in image intensity associated only with changes in surface orientation and illumination intensity. The algorithm detects colour edges by examining a chromaticity space version of the image in which intensity is eliminated. The algorithm keeps all small colour changes, but thresholds out all large ones in a manner derived from Horn’s version (Horn 1974) of Land’s retinex computation (Land 1977). The algorithm then reconstructs the image created by all the small changes which then encodes the shading information with the confounding surface-colour

information removed. The resulting shading field can be then used by any shape from shading algorithm to calculate surface shape.

A different approach to shape and colour is to consider whether or not colour information can be used directly to extract information about surface shape. Petrov (Petrov 1993) shows that if the illumination on a surface varies in colour as a function of incident angle that a given shape can be identified. He considers the set of possible images that could be produced under all illumination conditions. He then proves that, for uniformly curved surfaces, the set of colour images taken under all possible illuminants and painted any possible colour, forms a manifold on which a co-ordinate system can be introduced. A given shape will have specific coordinates in this space and so can be identified.

Another approach to shape and colour follows from an extension to the photometric stereo (Woodham 1980) method. In the original photometric stereo method, multiple images are taken under 3 different light sources. Given the images, positions and relative intensities of the sources, the surface normals at all image points can be calculated. In practice, photometric stereo is often implemented by using a single colour image with a red, green and blue light sources in different locations. The 3 bands of the colour image then have the same effect as 3 images taken in sequence.

The requirement that the light source positions and intensities be known can be relaxed as shown by Woodham (Woodham 1994). He considers the set of colours which might appear on an arbitrarily shaped object due to the fact that the colour of the incident light is varying as a function of its incident angle. For a sphere, which contains all possible orientations, he shows that the set of colours must fall on an ellipsoid in *RGB* colour space. This is analogous to the colours created by specularities and interreflections, discussed above, but now an ellipsoid is created instead of plane. An object with fewer orientations than a sphere will simply create part of the full ellipsoid. Different illumination conditions will result in different ellipsoids, so by fitting an ellipsoid to the available data, Woodham can extract the relevant parameters of the illumination. Once the illumination parameters are fixed, the regular photometric stereo method can be applied to determine the surface normals (up to a global rotation factor) at all locations.

Petrov's and Woodham's methods show that colour encodes shape information that can at least in principle be extracted. Their methods involve many restrictions and assumptions; nonetheless, it is interesting that colour has the potential to directly inform us about shape.

### **Colour and Texture**

Colour can aid the analysis of texture as has been shown in the work of Panjwani and Healey (Panjwani and Healey 1995). They developed a segmentation algorithm which segments images into interesting regions based on their colour texture. Figure 4 shows an example of the algorithm's excellent results.



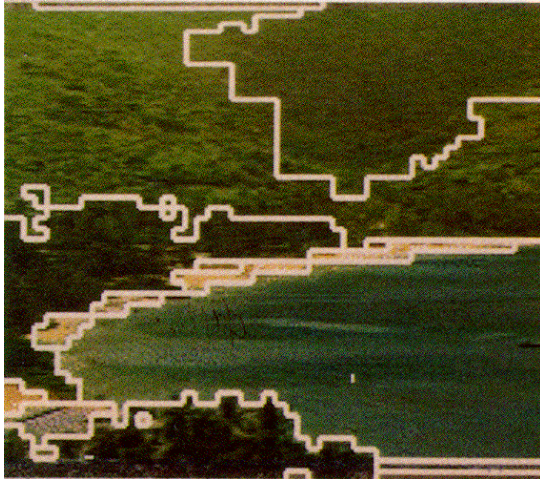


Figure 4: Top: Input image containing natural colour textures. Bottom: Image segmentation results based on aggregating similar colour textures.

Texture is a random pattern but with fixed properties, so the authors used Markov random fields to model the stochastic properties of a texture. In the case of texture in grey-level images, the Markov fields describe the statistical dependence of the intensity of a pixel as a function of the intensity of neighbouring pixels. In coloured textures, however, a statistical relationship also holds between the different colour channels, information lost by reduction to a single grey-level intensity channel. To exploit the local inter-dependencies between the colour planes, Panjwani and Healey use three-dimensional Markov fields to encode the stochastic properties of a texture not only within a colour plane, but also across colour planes. The Markov fields are used to determine a stochastic dependence between the *RGB* value of a pixel at a certain location and the *RGB* values at neighbouring locations. This model takes into account the dependence between all three colour planes and provides a compact description of a texture's properties.

Using this texture description method, scenes can be segmented based on the texture description obtained at each point in the scene. The textures in a scene differ according to the constraints embodied in the stochastic model with similar textures having similar Markov random field descriptors. Thus, similar pixels within a region of a single texture will belong to the same class and will be segmented as a single object. The segmentation algorithm is unsupervised (i.e., it makes no *a priori* assumptions regarding the textures present in a scene)

and operates in two stages. First is a region-splitting phase that divides the image into small regions that are guaranteed to contain only one type of texture. This is followed by a clustering phase that merges similar adjacent regions together until a stopping criterion is satisfied. After clustering, the textural properties of the resulting regions are recomputed.

As the results in Figure 4 show, the method is very good at creating an intuitively appealing segmentation. For example, the ocean has been appropriately segmented as a single region even though it contains a variety of different colours.

### **Other Applications of Colour**

In addition to the ways in which colour can help in the interpretation of images that have been discussed in detail in the preceding sections, there are numerous other ways in which colour has proven useful in computer vision. These applications of colour will be briefly described below.

Classifying the materials of surfaces in an image is another area in which colour can be helpful. Healey (Healey 1989) designed an algorithm that distinguishes between metals and inhomogeneous dielectrics by using the colour reflectance properties of these classes. The algorithm is founded on the observation that for metals the sensors measurements lie in a one-dimensional space (a line in the *RGB* colour space); whereas, in the case of inhomogeneous dielectrics, for which the dichromatic model of reflection applies, the sensor responses lie in a two-dimensional space (a parallelogram in the *RGB* colour space). The method, therefore, examines the colour-space histograms of the colours found in a region and uses the dimension of the space to determine the material type.

Colour also plays a role in the case of stereo correspondence. Jordan *et al.* (Jordan, Geisler and Bovik 1990) have shown that, in some cases, chromatic information is used more efficiently than luminance information. This provides some clues into the human stereo processing and how colour influences the visual system. Nguyen (Nguyen 1992) uses colour for stereo correspondence matching in a computer system.

Wet surfaces look different from dry ones. Mall *et al.* (Mall and Vitoria Lobo 1995) developed a model which predicts how the grey-level image of a dry surface changes when it is

wetted. Their model predicts the fact that wet surfaces are darker than dry ones and that the darkening caused by wetting is not the same as the simple darkening created by a shadow. Instead of being a simple uniform darkening as would occur in a shadow, the darkening of wetness is a function of the surface albedo. While Mall *et al.* applied their model to grey-level images only, for coloured surfaces the albedo varies with wavelength, so wetting will create colour shifts. It has yet to be tested, but it follows that colour should help in recognising surface wetness.

### **Colour Cues for Foveation**

Camera sensors usually maintain constant spatial resolution across the entire image. This differs from the human visual system, where the acuity is highly non-uniform and decreases from the optical axis to the periphery. Whether in a human or computer vision system (e.g. Funt 1980), varying resolution imagery necessitates saccadic movement of the eyes or camera to focus in on areas of interest to extract high resolution information. Swain et al. (Swain, Kahn and Ballard 1992) propose using simple multi-coloured regions as landmarks to guide the fixation process.

One aspect of the colour indexing method described above in the section “Colour-Based Object Recognition” is called *histogram backprojection*. It is a technique which finds any instances of a target object in an image based on the target’s colour histogram. To guide fixations, they suggest creating colour histograms of a few colourful regions in the visual field that can be used as landmarks. The regions do not have to correspond to an actual object, but rather are simple collections of colours. After a saccade, the landmarks can be found in the new view by histogram backprojection. Conveniently, colour histograms are not affected very much by changes in spatial resolution, which means that histogram created from the high resolution fovea can be used to backproject onto the low resolution periphery.

### **Is colour constancy really necessary?**

Many colour-based algorithms, such as ‘colour constant colour indexing’ or segmentation algorithms, require only relative colours to perform well and the computation of absolute colours (i.e. object surface colours as seen under a fixed canonical illuminant) is not necessary. In other cases, such as colour reproduction or digital photography, it is important to estimate absolute values for the colours in a scene. Often, there is more than one illuminant in a scene, so the chromaticity of a surface might vary spatially, which will affect most segmentation and object recognition algorithms. Some colour constancy algorithms (Barnard, Finlayson and Funt 1997) can eliminate the effect of multiple illuminants.

Colour constancy is an under-determined problem, so the solution will not be unique. It has been shown that under certain circumstances (Finlayson, Drew and Funt, 1994a) images can be converted to make them look like they were taken under a standard ‘canonical’ illuminant. This transformation is done by a simple diagonal transformation for all the pixels in a scene. Thus, once the colours of the initial and canonical illuminants are specified, the problem is reduced to estimating the colour of the illuminant.

We designed a neural network (Funt, Cardei and Barnard 1996; Cardei, Funt and Barnard 1997) that is able to recover the chromaticity of the illuminant in a scene, based only on a binarised chromaticity histogram of the image. The neural network is trained on a large set of synthesised scenes for which the illuminant is known. Once the learning period is completed, the network is presented a histogram of a scene and then outputs an estimate of the chromaticity of the illuminant. The estimates obtained are comparable with the best colour constancy algorithms to date (e.g. Finlayson 1996 and 1997, Forsyth 1990) and much superior to traditional methods such as retinex (Land 1977). This non-parametric approach to colour constancy has the advantage that it has no built-in constraints and that it is adaptive and can always be retrained for new conditions such as a different cameras. An example of the colour correction carried out by the neural network is shown in Figure 5.



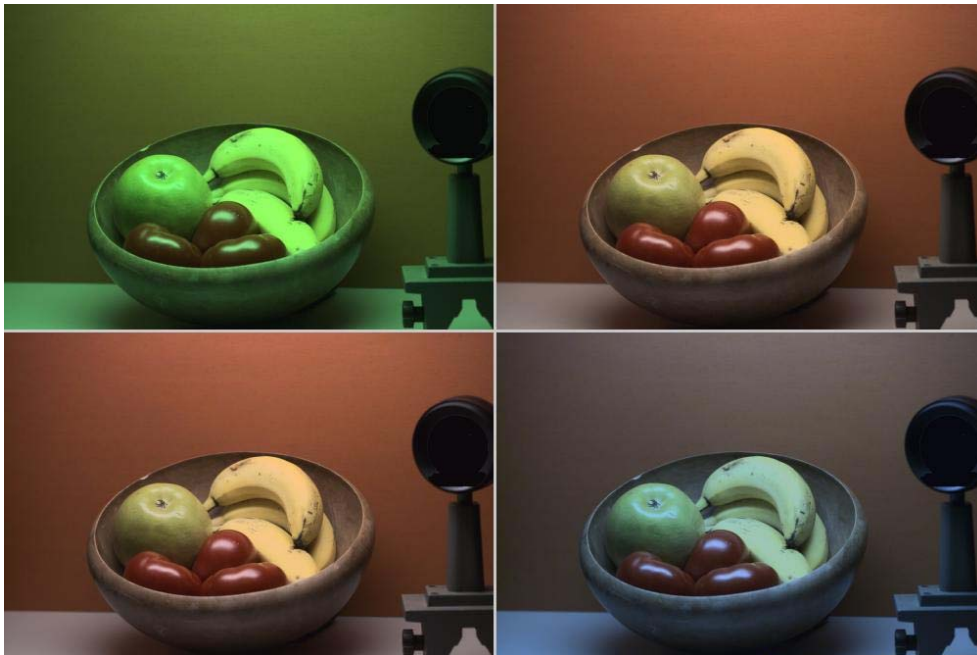


Figure 5. An example of neural network based colour correction results. Top left: input image taken under unknown illumination. Top right: target image taken under the canonical illumination. Bottom Left: good colour correction result obtained based on neural network estimation of the illumination. Bottom right: poor colour correction result obtained based on grey world estimation of the illumination

## Conclusion

Whether or not a computer vision system employs colour imagery, the world contains colours. Reducing the colours to one-dimensional grey level images is perhaps more likely to make the image interpretation more rather than less difficult. We have tried to demonstrate this point by example and have consider how colour can be used in computer vision systems to help in interpreting interreflections, specularities, material types, and surface wetness as well as in recognising objects, enhancing stereo correspondences, providing foveation landmarks and extracting shape. Many other applications of colour are possible, interpreting shadows or recognising transparency, for example.

We find it somewhat surprising that none of these example applications require absolute colour, that is object surface colour which is independent of the illuminant. For all of them, changes in the illumination either do not matter or can easily be factor out by considering colour

ratios. While humans clearly possess a fair degree of colour constancy and colour constancy can also be achieved in a computer vision system, it remains unclear exactly what crucial role it might play.

### **Acknowledgement**

This work has been supported by the Natural Sciences and Engineering Research Council of Canada.

### **References**

- Barnard, K., G. Finlayson and B. Funt (1997). Color Constancy for scenes with varying illumination. *Computer Vision and Image Understanding*, 65(2), 311-321.
- Brainard, D. H., W.A. Brunt, J.M. Speigle (1997). Color constancy in the nearly natural image. 1. Asymmetric matches. *J. Opt. Soc. Am. A*, 14(9).
- Cardei, V., B. Funt and K. Barnard (1997). Modeling Color Constancy with Neural Networks. *Proc. Int. Conf. on Vision, Recognition, and Action: Neural Models of Mind and Machine*, Boston, May 29-31.
- Finlayson G., P. M. Hubel and S. Hordley (1997). Color by Correlation. *Proc. IS&T/SID Fifth Color Imaging Conference: Color Science, Systems and Applications*, 6-11.
- Finlayson G. (1996). Color in Perspective. *IEEE Trans. on PAMI*, Vol. 18 , No.10, 1034-1038.
- Finlayson, G., M. Drew, and B. Funt (1994a). Color Constancy: Generalized Diagonal Transforms Suffice. *J. Opt. Soc. Am. A*, 11(11), 3011-3020.
- Finlayson, G., M. Drew, and B. Funt (1994b). Spectral Sharpening: Sensor Transformations for Improved Color Constancy. *J. Opt. Soc. Am. A*, 11(5), 1553-1563.
- Forsyth D.A. (1990). A Novel Algorithm for Color Constancy. *Int. J. of Computer Vision*, 5:1, 5-36.
- Funt, B. (1980). Problem-Solving with Diagrammatic Representations. *Artificial Intelligence*, Vol. 13, 201-230.
- Funt, B., M. Drew, and M. Brockington (1992). Recovering Shading from Color Images. *ECCV'92*, 124-132.
- Funt, B., and M. Drew (1993). Color Space Analysis of Mutual Illumination. *IEEE Trans. on PAMI*, Vol. 15, No. 12.

Appears in *Color Perception: Philosophical, Psychological, Artistic and Computational Perspectives*, *Vancouver Studies in Cognitive Science* Vol 9, ed. S. Davis, Oxford University Press, 2000. Copyright OUP.

- Funt, B., and G. Finlayson (1995). Color Constant Color Indexing. *IEEE Trans. on PAMI*, Vol. 17, No. 5.
- Funt, B., V. Cardei, and K. Barnard (1996). Learning Color Constancy. *Proc. IS&T/SID Fourth Color Imaging Conference: Color Science, Systems and Applications*, 58-60.
- Healey, G. (1989). Using Color for Geometry-Insensitive Segmentation. *J. Opt. Soc. Am. A*, Vol.6, No.6, 920-936.
- Horn, B.K.P (1974). Determining Lightness from an Image. *Computer Graphics and Image Processing*, Vol. 3, No. 1, 277-299.
- Jordan III, J.R., W. Geisler, and A. Bovik (1990). Color as a Source of Information in the Stereo Correspondence Process. *Vision Res.* Vol. 30, No. 12, 1955-1970.
- Land, E.H. (1977). The Retinex Theory of Color Vision. *Scientific American*, 108-129.
- Mall H., N. da Vitoria Lobo (1995). Determining Wet Surfaces Versus Dry, *Proc. ICCV'95 Fifth International Conference on Computer Vision*, IEEE Computer Society Press, pp. 963-968, June 1995.
- Nguyen, H.H., Cohen, P. (1992). Correspondence from Color Shading, *ICPR92 Proc. Intl. Conf. On Pattern Recognition (I:113-116)*, 1992.
- Panjwani, D.K., and G. Healy (1995). Markov Random Field Models for Unsupervised Segmentation of Textured Color Images. *IEEE Trans. on PAMI*, Vol. 17, No. 10.
- Petrov, A.P. (1993). On Obtaining Shape from Color Shading. *Color Res. and Appl.*, Vol. 18, No.6, 375-379.
- Shafer S.A. (1985). Using color to separate reflection components. *Color Res. Appl.*, Vol.10, No. 4, 210-218.
- Swain, M., and D. Ballard (1991). Color Indexing. *Int. J. of Computer Vision*, 7:1, 11-32.
- Swain, M., R.E. Kahn and D. Ballard (1992). Low Resolution Cues for Guiding Saccadic Eye Movements. *Proc. CVPR IEEE Computer Vision and Pattern Recognition Conference*, 737-740.
- Woodham, R.J. (1980). Photometric Method for Determining Surface Orientation from Multiple Images. *Optical Engineering*, Vol. 19. No. 1, 139-144.
- Woodham, R.J. (1994). Gradient and Curvature from the Photometric-Stereo Method, Including Local Confidence Estimation. *J. Opt. Soc. of Am. A*, Vol. 11(11), 3050-3068.