Color Constancy

Marc Ebner, Ernst Moritz Arndt Universität Greifswald

Related Concepts

– Automatic White Balance

Definition

Color constancy is the ability to perceive colors as approximately constant even though the light entering the eye varies with the illuminant. Color constancy also names the field of research investigating the extent of this ability, i.e. the conditions under which a color is actually perceived as constant and which factors influence color constancy. Computer scientists working in the field of color constancy try to mimic this ability in order to produce images which are independent of the illuminant, i.e. color constant. Simple color constancy algorithms, also known under the name automatic white balance, are used in digital cameras to compute a color corrected output image. The input of a color constancy algorithm is often one image taken under an arbitrary illuminant and the output of a color constancy algorithm is frequently the image as it would appear had it been taken with a canonical illuminant such as CIE Standard Illuminant D65 or a spectrally uniform illuminant.

Background

An introduction to computational color constancy is given by Ebner [1]. Maloney [2] also reviews different algorithms for surface color perception. Color is a product of the brain [3]. When an observer perceives an object, processing starts with the retinal sensors. The sensors in the retina measure the light entering the eye. However, the light entering the eye is dependent on both the spectral characteristics of the illuminant and the reflectance properties of the object. Therefore, without any additional processing, the measured light varies with the illuminant.

In the eye, two different types of retinal sensors exist: rods and cones. The rods are used for viewing when little light is available. The cones are mostly used in bright light conditions for color vision. Three types of cones can be distinguished which absorb light primarily in the red, green, and blue parts of the spectrum. Similarly, a digital sensor often measures the incident light in three different parts of the spectrum and uses red, green and blue sub-sensors. However, cameras with four sub-sensors, e.g. red, green, blue and cyan also exist.

Suppose that an observer views a diffusely reflecting surface which uniformly reflects the incident light. Now assume that the surface is illuminated by a candle. A candle emits more light towards the red part of the spectrum. The candle light will reach the surface where part of the light will be absorbed and the remainder will be reflected. Part of the reflected light enters the eye where it is measured. The sensitivity function of the sensors in combination with the amount of light entering the eye will determine how strongly the sensors respond. Now consider another illuminant with a higher color temperature, e.g. daylight or an electronic flash. Such an illuminant will emit more light towards the blue spectrum compared to the candle. If the same surface is viewed with an illuminant that has a high color temperature then the sensors shift their response towards the blue part of the spectrum. Assuming a normalized response of all three sensors, then the reflected light will have a red color cast when the surface is illuminated by a candle. The measured light will appear white during daylight and it will have a blueish color cast for an illuminant with a high color temperature. The observer, however will be able to call out the correct color of the surface independent of the illuminant. Color constancy algorithms try to mimic this ability by computing an image which is independent of the illuminant.

Theory

Let $I(\lambda, x, y)$ be the irradiance captured by either a digital sensor or by the eye at position (x, y) for wavelength λ. Let $S_i(\lambda)$ be the response function of sensor i. Then the response of the sensor $c_i(x, y)$ at position (x, y) is given by

$$c_i(x, y) = \int S_i(\lambda) I(\lambda, x, y) d\lambda.$$

The integration is done over all wavelengths to which the sensor responds. Assuming three receptors with sensitivity in the red, green and blue parts of the spectrum, then $i \in \{r, g, b\}$. In this case, the measurement of the sensor is a three component vector $\mathbf{c} = [c_r, c_g, c_b]$.

The irradiance I falling onto the sensor is a result of the light reflected from an object patch. Let $L(\lambda, x, y)$ be the irradiance falling onto a diffusely reflecting object patch which is imaged at position (x, y) of the sensor arrangement. Let $R(\lambda, x, y)$ be the reflectance of the imaged object patch. Thus,

$$I(\lambda, x, y) = G(x, y) R(\lambda, x, y) L(\lambda, x, y)$$

where $G(x, y)$ is a geometry factor which takes the orientation between the surface and the light source into account. For a diffusely reflecting surface, $G(x, y) = \cos\alpha$ where α is the angle between the unit vector which points from the surface into the direction of the light source and the normal vector at the corresponding surface position. Thus, the sensor response can be written as

$$c_i(x, y) = G(x, y) \int S_i(\lambda) R(\lambda, x, y) L(\lambda, x, y) d\lambda.$$

From this equation, it is apparent that the sensor response depends on the orientation of the patch relative to the light source (because of $G(x, y)$), it depends on the sensitivity S_i of the sensor i, the reflectance of the object patch $R(\lambda, x, y)$ and on the illuminant $L(\lambda, x, y)$. Some color constancy algorithms are based on a set of basis functions to model illuminants and reflectances. See Maloney [2] for an introduction.

Color constancy algorithms frequently assume that the sensitivity of the sensors is very narrow band. Assuming that they have the shape of a delta function, $S_i = \delta(\lambda - \lambda_i)$, it holds that

$$c_i(x, y) = G(x, y) \int \delta(\lambda - \lambda_i) R(\lambda, x, y) L(\lambda, x, y) d\lambda,$$
$$c_i(x, y) = G(x, y) R(\lambda_i, x, y) L(\lambda_i, x, y).$$

This equation is often written in the form

$$c_i(x, y) = G(x, y) R_i(x, y) L_i(x, y)$$

where the only difference to the previous equation is that the index i is used instead of the parameter λ_i. In this treatment, sensor response, reflectance and the illuminant is considered only for three distinct wavelengths, i.e. color bands, with $i \in \{r, g, b\}$.

A color constancy algorithm tries to discount the illuminant by computing a color constant descriptor $d(\mathbf{c})$ which is independent of the illuminant $\mathbf{L}(x, y) = [L_r(x, y), L_g(x, y), L_b(x, y)]$. Geusebroek et al. [4] have derived several different descriptors which can be computed from $\mathbf{c}$ and are invariant to some imaging conditions such as viewing direction, surface orientation, highlights, illumination direction, illumination intensity or illumination color. Finlayson and Hordley [5] have shown how a color constant descriptor can be computed provided that the illuminant can be approximated by a black body radiator. Apart from computing a color constant descriptor a color constancy algorithm usually tries to output an image of the scene which would either correspond to the perception of a human photographer observing the scene or it would correspond to the image that would have resulted if a spectrally uniform illuminant or illuminant D65 had been used.

An ideal solution to the problem of color constancy would be to compute $\mathbf{R}(x, y) = [R_r(x, y), R_g(x, y), R_b(x, y)]$ from the sensor responses. It is of course clear that this problem cannot be solved without making additional assumptions, because for each position on the sensor array, one only has three measurements but there are seven unknowns (shading, reflectance and illumination components). Note that the above model for image generation is already a simple model assuming narrow band sensor responses.

A frequently made assumption is that the illuminant varies slowly over the image while reflectance is able to change abruptly between sensor responses. Since color constancy algorithms are based on certain assumptions, they will not work correctly if the assumptions are violated. In many cases, it is possible to find images where the color constancy algorithm does not perform as intended. The goal is to develop algorithms which perform well on most everyday scenes.

Simple Algorithms If a single illuminant illuminates the scene uniformly, i.e. $L_i(x, y) = L_i$, then it suffices to estimate a three component vector $\tilde{\mathbf{L}}$ from all the measured sensor responses with $\tilde{\mathbf{L}} \approx \mathbf{L}$. Given an estimate $\tilde{\mathbf{L}}(x, y)$, a color constant descriptor can be computed by dividing the sensor response by the estimate of the illuminant.

$$\frac{c_i(x,y)}{\tilde{L}_i(x,y)} = \frac{G(x,y)R_i(x,y)L_i(x,y)}{\tilde{L}_i(x,y)} \approx \frac{G(x,y)R_i(x,y)L_i(x,y)}{L_i(x,y)} = G(x,y)R_i(x,y)$$

Such an output image will be a shaded reflectance image. In other words, a diagonal color transform suffices if the sensor response is very narrow band. Some color constancy algorithms, however, also use a general 3×3 matrix transform to compute a color corrected output image.

It is also possible to transform a given image taken under one illuminant $\mathbf{L}$ to another image taken under a different illuminant $\mathbf{L}'$. This can be done by multiplying each sensor response vector by a diagonal matrix whose elements are set to $k_i = \frac{L'_i}{L_i}$. The coefficients k_i are called von Kries coefficients [6]. Necessary and sufficient conditions on whether von Kries chromatic adaptation gives color constancy have been derived by West and Brill [7].

A simple algorithm to estimate the color of the illuminant is the white patch Retinex algorithm. It is a simplified version of the parallel Retinex algorithm [8]. In order to understand how this algorithm works, suppose that a white patch, i.e. a uniformly reflecting patch, is contained in the imaged scene which is uniformly illuminated. Assuming a normalized sensor response, then the response of the sensors on the white patch will be an estimate of the illuminant. For the white patch, it holds that $R_i = 1$ which leads to $c_i(\text{white patch}) = GL_i$. The white patch algorithm treats each color band separately and searches for the maximum response which is assumed to be an estimate of the illuminant.

$$\tilde{L}_i = \max_{x,y} c_i(x,y)$$

Instead of locating the maximum response, one can also compute a histogram of the sensor responses and then set the estimate of the illuminant at some percentage from above. This will lead to a more robust estimate of the illuminant. Figure 1(b) shows the output of the white patch Retinex algorithm for a sample image shown in Figure 1(a). The illuminant was estimated at 5% from above using a histogram approach for each color band.

Another simple algorithm is based on the gray world assumption which is due to Buchsbaum [9]. According to Buchsbaum, the world is gray on average. Let a_i be the global average of all sensor responses for color channel i where n is the number of sensors.

$$a_i = \frac{1}{n}\sum_{x,y} c_i(x,y) = \frac{1}{n}\sum_{x,y} G(x,y)R_i(x,y)L_i(x,y)$$

Assuming a uniform illuminant $L_i(x,y) = L_i$ and an independence between shading and reflectance, then

$$a_i = L_i\frac{1}{n}\sum_{x,y} G(x,y)R_i(x,y) = L_iE[G(x,y)R_i(x,y)] = L_iE[G(x,y)][R_i(x,y)]$$

(a) (b) (c)

Fig. 1. (a) sample input image (b) results for the white patch Retinex algorithm using a histogram (c) results for the gray world assumption.

where $E[x]$ denotes the expected value of x. Suppose that a large number of differently colored objects are contained in the scene. Thus, a uniform distribution of colors is assumed. This results in $E[R_i(x,y)] = \frac{1}{n}\sum_{x,y} R_i(x,y) = \frac{1}{2}$ assuming a range of $[0,1]$ for reflectances and $E[G(x,y)] = c$ where c is a constant. The result is

$$a_i = \frac{c}{2} L_i.$$

Hence, the illuminant is proportional to global space average color $L_i \propto a_i$ and an estimate of the illuminant can be obtained by setting

$$\tilde{L}_i = 2a_i.$$

Using $c = 1$ assumes that all patches are frontally oriented or alternatively, that the geometry factor is subsumed into a combined reflectance and geometry factor. Figure 1(c) shows the results for the gray world assumption on a sample image. Figure 2 shows the results for another image where the assumption, that a large number of different colored objects are contained in the scene, is not fulfilled. In this case, the gray world assumption will not work correctly.

Additional Color Constancy Algorithms Other important algorithms include gamut constraint algorithms originally developed by Forsyth [10]. A gamut constraint algorithm looks at the gamut of colors contained in a sample image

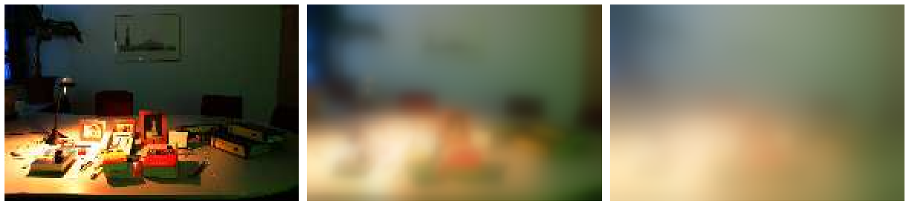
(a) (b)

Fig. 2. (a) image of a leaf from a banana plant (b) results for the gray world assumption.

and then transforms this color gamut to a color gamut of a canonical image. Forsyth's algorithm operates on three color channels. Finlayson [11] has developed a variant which operates with a projected color gamut (2D gamut constraint algorithm). Van de Weijer et al. [12] have introduced the gray-edge hypothesis. While the gray-world assumption suggests that the world is gray on average, the gray-edge hypothesis suggests that image derivatives are gray on average. Brainard and Freeman [13] have addressed the problem of color constancy using Bayesian decision theory.

Uniform vs non-uniform illumination Many color constancy algorithms assume that the scene is uniformly illuminated by a single illuminant. If multiple light sources are distributed over the scene, then it is assumed that these light sources can be combined into a single average illuminant. This is possible provided that the light sources are sufficiently distant from the scene. If the illumination is uniform, then only a three component descriptor has to be estimated from the input image. However, in practice, many scenes are illuminated non-uniformly. Very often, one has several different illuminants. For instance, daylight may be falling through a window while artificial lights have already been turned on inside a building. Non-uniform illumination may also be present outside during a sunny day. Consider a family sitting in the garden under a red umbrella and a photographer taking a photograph of the family members. The family would be illuminated by light reflected from the red umbrella while the surrounding would be illuminated by direct sunlight. A digital camera usually corrects for a single illuminant. Thus, either the family members would have a red color cast to them or the background colors would not look right in the resulting image.

Algorithms have also been developed which can cope with a spatially varying illuminant. Land and McCann's Retinex algorithm [8] is a parallel algorithm for color constancy which allows for a non-uniform illumination. They only considered one-dimensional paths over the sensor array. Horn [14] extended Land's Retinex algorithm to two dimensions. Blake [15] provided additional improve-



0 200 2000

Fig. 3. Local space average color after 0, 200 and 2000 iterations.

ments. Barnard et al. [16] extended the 2D gamut constraint algorithm to scenes with varying illumination.

Ebner [1] showed how a grid of processing elements is able to estimate the color of the illuminant locally using the gray world assumption. Each processing element receives the measurement c_i from the sensor and computes local space average color. The processing elements are laterally connected to neighboring processing elements. Let $a_i(x, y)$ be the current estimate of local space average color for a processing element located at position (x, y). Each processing element receives the estimate of local space average color from neighboring elements and averages the neighboring estimates to update its own estimate. A small component from the sensor measurement is then added to this estimate. Let p be a small percentage and let $N(x, y)$ be the neighborhood of the processing element at position (x, y) then the computation of a processing element consists of the following two updates.

$$a'_i(x, y) = \frac{1}{|N(x, y)|} \sum_{(x',y')\in N(x,y)} a_i(x', y')$$
$$a_i(x, y) = (1 - p)a'_i(x, y) + pc_i(x, y)$$

The two updates are carried out iteratively. This process converges to local space average color which is an estimate of the illuminant, $\tilde{L}_i(x, y) = 2a_i(x, y)$. Figure 3 shows local space average color after 0, 200 and 2000 iterations using $p = 0.0001$ given the input image shown in Figure 4(b). The extent of the averaging is determined by the parameter p. Figure 4 shows the output when local space average color is used to estimate the illuminant locally. Local average color was computed for a down-scaled image (25% in each direction) the original image had 768×512 pixels. Ebner [1] suggested that a similar algorithm may used by the brain for color perception.

Advanced reflectance models The theoretical model of color image formation, that has been given above, assumed that objects diffusely reflect the incident light. This is not the case for all surfaces, e.g. brushed metal or pastics. Especially for plastic objects or objects covered with gloss varnish a more elaborate reflectance model is more appropriate. The dichromatic reflection model [17,18]

(a) (b)

(c) (d)

Fig. 4. (a & b) input images (c & d) results using local space average color.

assumes that reflectance is composed of interface reflectance, which occurs at the boundary between the object's surface and air, and body reflectance which is due to the scattering of light below the object's surface. In other words, the reflection of light from the object's surface is assumed to be partially specular and partially diffuse. Color constancy algorithms have also been developed using the dichromatic reflection model, e.g. by Risson [19].

Application

Color constancy algorithms are ideal for color correction in digital photography. In digital photography, the goal is to obtain a color corrected image that corresponds nicely to human perception. A printed photograph or a photograph viewed on a computer display should appear in exactly the same way that the human observer (the photographer) perceived the scene. Besides digital photography, color constancy algorithms can be applied in the context of most computer vision tasks. For many tasks one should try to estimate object reflectances. For instance, image segmentation would not be as difficult, if object reflectance could be correctly determined. Similarly, color-based object recognition is easier if performed on reflectance information. Thus, color constancy algorithms should often be applied as a pre-processing step. This holds especially for autonomous mobile robots equipped with a vision system because autonomous mobile robots need to operate in different environments under different illuminants.

Experimental Results and Datasets

A comparison of computational color constancy algorithms is given by Barnard et al. [20] for synthetic as well as real image data. Another detailed comparison of color constancy algorithms along with pseudo code for many algorithms is given by Ebner [1]. Data for computer vision and computational color science can be found at the Simon Frasier University, Canada (`www.cs.sfu.ca/~colour/data/`). A repository has also been created by the color group at the University of East Anglia, UK (`www.colour-research.com`). A database for spectral color science is available from the University of Eastern Finland, Finland (`spectral.joensuu.fi/`).

Recommended Readings

[1] Ebner, M. (2007). Color Constancy. John Wiley & Sons, England

[2] Maloney, L.T. (1999). Physics-based approaches to modeling surface color perception. In Gegenfurtner, K.R., Sharpe, L.T., eds.: Color Vision: From Genes to Perception, Cambridge, UK, Cambridge University Press 387–422

[3] Zeki, S. (1993). A Vision of the Brain. Blackwell Science, Oxford

[4] Geusebroek, J.M., van den Boomgaard, R., Smeulders, A.W.M., Geerts, H. (2001). Color invariance. IEEE Transactions on Pattern Analysis and Machine Intelligence **23**(12) 1338–1350

[5] Finlayson, G.D., Hordley, S.D. (2001). Color constancy at a pixel. Journal of the Optical Society of America A **18**(2) 253–264

[6] Richards, W., Parks, E.A. (1971). Model for color conversion. Journal of the Optical Society of America **61**(7) 971–976

[7] West, G., Brill, M.H. (1982). Necessary and sufficient conditions for von Kries chromatic adaptation to give color constancy. Journal of Mathematical Biology **15** 249–258

[8] Land, E.H., McCann, J.J. (1971). Lightness and retinex theory. Journal of the Optical Society of America **61**(1) 1–11

[9] Buchsbaum, G. (1980). A spatial processor model for object colour perception. Journal of the Franklin Institute **310**(1) 337–350

[10] Forsyth, D.A. (1990). A novel algorithm for color constancy. International Journal of Computer Vision **5**(1) 5–36

[11] Finlayson, G.D. (1996). Color in perspective. IEEE Transactions on Pattern Analysis and Machine Intelligence **18**(10) 1034–1038

[12] van de Weijer, J., Gevers, T., Gijsenij, A. (2007). Edge-based color constancy. IEEE Transactions on Image Processing

[13] Brainard, D.H., Freeman, W.T. (1997). Bayesian color constancy. Journal of the Optical Society of America A **14**(7) 1393–1411

[14] Horn, B.K.P. (1974). Determining lightness from an image. Computer Graphics and Image Processing **3** 277–299

[15] Blake, A. (1985). Boundary conditions for lightness computation in mondrian world. Computer Vision, Graphics, and Image Processing **32** 314–327

[16] Barnard, K., Finlayson, G., Funt, B. (1997). Color constancy for scenes with varying illumination. Computer Vision and Image Understanding **65**(2) 311–321

[17] Klinker, G.J., Shafer, S.A., Kanade, T. (1988). The measurement of highlights in color images. International Journal of Computer Vision **2** 7–32

[18] Tominaga, S. (1991). Surface identification using the dichromatic reflection model. IEEE Transactions on Pattern Analysis and Machine Intelligence **13**(7) 658–670

[19] Risson, V.J. (2003). Determination of an illuminant of digital color image by segmentation and filtering. United States Patent Application, Pub. No. US 2003/0095704 A1

[20] Barnard, K., Cardei, V., Funt, B. (2002). A comparison of computational color constancy algorithms – part I and II. IEEE Transactions on Image Processing **11**(9) 972–996