

Color Constancy Using Local Color Shifts

Marc Ebner

Universität Würzburg, Lehrstuhl für Informatik II,
Am Hubland, 97074 Würzburg, Germany,
ebner@informatik.uni-wuerzburg.de,
<http://www2.informatik.uni-wuerzburg.de/staff/ebner/welcome.html>

Abstract. The human visual system is able to correctly determine the color of objects in view irrespective of the illuminant. This ability to compute color constant descriptors is known as color constancy. We have developed a parallel algorithm for color constancy. This algorithm is based on the computation of local space average color using a grid of processing elements. We have one processing element per image pixel. Each processing element has access to the data stored in neighboring elements. Local space average color is used to shift the color of the input pixel in the direction of the gray vector. The computations are executed inside the unit color cube. The color of the input pixel as well as local space average color is simply a vector inside this Euclidean space. We compute the component of local space average color which is orthogonal to the gray vector. This component is subtracted from the color of the input pixel to compute a color corrected image. Before performing the color correction step we can also normalize both colors. In this case, the resulting color is rescaled to the original intensity of the input color such that the image brightness remains unchanged.

1 Motivation

The human visual system is able to correctly determine the color of objects irrespective of the light which illuminates the objects. For instance, if we are in a room illuminated with yellow lights, we are nevertheless able to determine the correct color of the objects inside the room. If the room has a white wall, it will reflect more red and green light compared to the light reflected in the blue spectrum. Still, we are able to determine that the color of the wall is white. However, if we take a photograph of the wall, it will look yellow. This occurs because a camera measures the light reflected by the object. The light reflected by the object can be approximated as being proportional to the amount of light illuminating the object and the reflectance of the object for any given wavelength. The reflectance of the object specifies the percentage of the incident light which is reflected by the object's surface. The human visual system is somehow able to discount the illuminant and to calculate color constant descriptors if the scene is sufficiently complex. This ability is called color constancy. In other words, the human visual system is able to estimate the actual reflectance, i.e. the color of

the object. The perceived color stays constant irrespective of the illuminant used to illuminate the scene.

Land, a pioneer in color constancy research, has developed the retinex theory [1,2]. Others have added to this research and have proposed variants of the retinex theory [3,4,5,6,7,8]. Algorithms for color constancy include gamut-constraint methods [9,10,11], perspective color constancy [12], color by correlation [13,14], the gray world assumption [15,16], recovery of basis function coefficients [17,18,19], mechanisms of light adaptation coupled with eye movements [20], neural networks [21,22,23,24,25,26], minimization of an energy function [27], comprehensive color normalization [28], committee-based methods which combine the output of several different color constancy algorithms [29] or use of genetic programming [30]. Risson [31] describes a method to determine the illuminant by image segmentation and filtering of regions which do not agree with an assumed color model.

We have developed a parallel algorithm for color constancy [32,33,34]. The algorithm computes local space average color using a parallel grid of processing elements. Note that local space average color is not the same as global space average color. Global space average color assumes a single illuminant. In contrast, we do not assume a uniform illumination of the scene. Local space average color is taken as an estimate of the illuminant for each image pixel. This estimate of the illuminant is then used to perform a local color correction step for each image pixel.

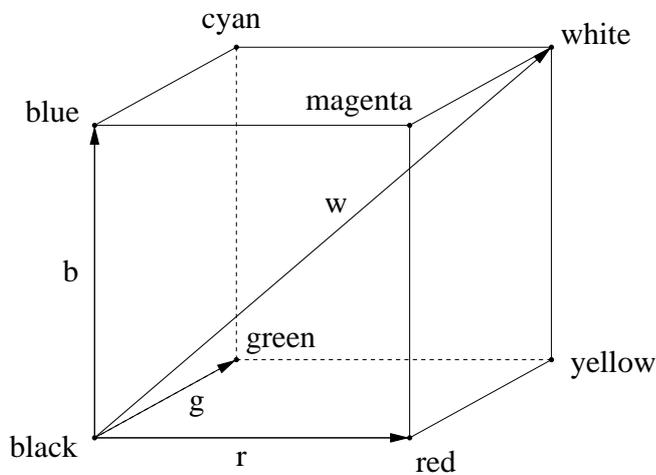


Fig. 1. RGB color space. The space is defined by the three vectors \mathbf{r} (red), \mathbf{g} (green), and \mathbf{b} (blue). The gray vector \mathbf{w} passes through the center of the cube from black to white.

2 RGB Color Space

Let us visualize the space of possible colors as a color cube [35,36] and let $\mathbf{r} = [1, 0, 0]^T$, $\mathbf{g} = [0, 1, 0]^T$, $\mathbf{b} = [0, 0, 1]^T$ be the three color vectors red, green and blue, which define the cube. The color components are normalized to the range $[0, 1]$. Therefore, all colors are located inside the unit cube. The eight corners of the cube can be labeled with the colors black, red, green, blue, magenta, cyan, yellow, and white. The gray vector passes through the cube from $[0, 0, 0]^T$ to $[1, 1, 1]^T$. This RGB color space is shown in Figure 1.

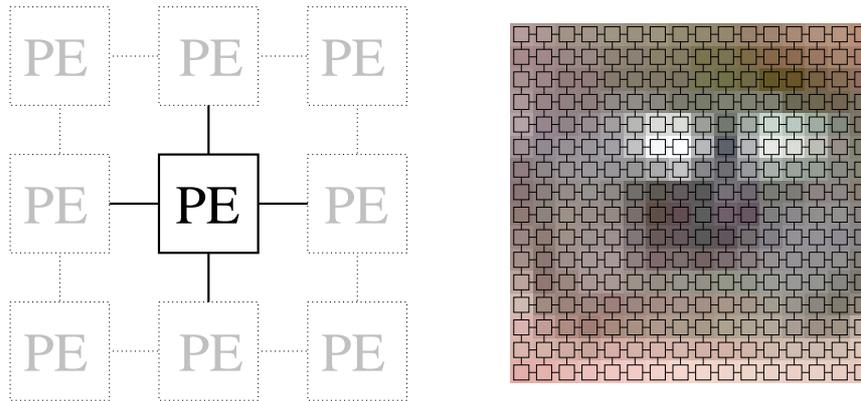


Fig. 2. Each processing element has access to information stored at neighboring processing elements (left). A matrix of processing elements with one processing element per pixel is used (right).

3 Parallel Computation of Local Space Average Color

The algorithm operates on a grid of processing elements. Each processing element has access to the color of a single image pixel. It also has access to data stored and computed by four neighboring processing elements (Figure 2). We have one processing element per image pixel. The algorithm first determines local space average color. Local space average color is calculated iteratively by averaging estimates of the local space average color from neighboring elements. Let $a_i(x, y)$ with $i \in \{r, g, b\}$ be the current estimate of the local space average color of channel i at position (x, y) in the image. Let $c_i(x, y)$ be the intensity of channel i at position (x, y) in the image. Let p be a small percentage greater than zero. We iterate the following two steps indefinitely.

- 1.) $a'_i(x, y) = (a_i(x - 1, y) + a_i(x + 1, y) + a_i(x, y - 1) + a_i(x, y + 1))/4.0$
- 2.) $a_i(x, y) = c_i(x, y) \cdot p + a'_i(x, y) \cdot (1 - p)$

The first step averages the estimate obtained from the neighboring elements on the left and right, as well as above and below. The second step slowly fades the color of the current pixel $c_i(x, y)$ into the current average. As a result, we obtain local space average color for each pixel of the image. Note that initialization of a_i can be arbitrary as it decays over time due to the multiplication by the factor $(1 - p)$. Figure 3 shows how local space average color is computed with this method for an input image. Since the intensities are averaged for each color channel, it is important that the input data is linear. If necessary, it must be linearized by applying a gamma correction.



Fig. 3. Local space average color is computed iteratively. The images show local space average color after 1, 50, 200, 1000, 5000, and 20000 steps. For this image 22893 steps were needed until convergence.

As the estimate of local space average color is handed from one element to the next, it is multiplied with the factor $(1 - p)$. Therefore, a pixel located n steps from the current pixel will only contribute with a factor of $(1 - p)^n$ to the new estimate of local space average color. The above computation is equivalent to the convolution of the input image with the function $e^{-\frac{|r|}{\sigma}}$ where $r = \sqrt{x^2 + y^2}$ is the distance from the current pixel and σ is a scaling factor. In this case, local space average color \mathbf{a} is given by

$$\mathbf{a} = k \int_{x,y} \mathbf{c} e^{-\frac{|r|}{\sigma}} dx dy \quad (1)$$

where k is chosen such that

$$k \int_{x,y} e^{-\frac{|r|}{\sigma}} dx dy = 1. \quad (2)$$

Thus, the factor p essentially determines the radius over which local space average color is computed. Figure 4 shows the result for different values of p . In practice, this type of computation can be performed using a resistive grid [7,25].



Fig. 4. The parameter p determines the extent over which local space average color will be computed. If p is large, then local space average color will be computed for a small area. If p is small, then local space average color will be computed for a large area.

Alternatively, instead of using a convolution with $e^{-\frac{|r|}{\sigma}}$ to compute local space average color, one can also compute space average color by convolving the input image with a Gaussian. In this case, local space average color is given by

$$\mathbf{a} = k \int_{x,y} \mathbf{c} e^{-\frac{r^2}{\sigma^2}} dx dy \quad (3)$$

where k is chosen such that

$$k \int_{x,y} e^{-\frac{r^2}{\sigma^2}} dx dy = 1. \quad (4)$$

This type of convolution is used by Rahman et al. [8] to perform color correction using several Gaussians of varying extent for each image pixel.

4 Color Constancy Using Color Shifts

According to the gray world hypothesis, on average, the color of the world is gray [15,16]. If space average color deviates from gray, it has to be corrected. Instead of rescaling the color channels, we use local space average color to shift the color vector in the direction of the gray vector. The distance between space average color and the gray vector determines how much local space average color deviates from the assumption that, on average, the world is gray. Let $\mathbf{w} = \frac{1}{\sqrt{3}}[1, 1, 1]^T$ be the normalized gray vector. Let $\mathbf{c} = [c_r, c_g, c_b]^T$ be the color of the current pixel and $\mathbf{a} = [a_r, a_g, a_b]^T$ be local space average color. We first project local space average color onto the gray vector. The result is subtracted from local space average color. This gives us a vector which is orthogonal to the gray vector. It

points from the gray vector to the local space average color. Its length is equal to the distance between local space average color and the gray vector.

$$\mathbf{a}_\perp = \mathbf{a} - (\mathbf{a} \cdot \mathbf{w})\mathbf{w} \quad (5)$$

This vector is subtracted from the color of the current pixel in order to undo the color change due to the illuminant. The output color is calculated as

$$\mathbf{o} = \mathbf{c} - \mathbf{a}_\perp \quad (6)$$

or

$$o_i = c_i - a_i + \frac{1}{3}(a_r + a_g + a_b) \quad (7)$$

where $i \in \{r, g, b\}$. If we define $\bar{a} = \frac{1}{3}(a_r + a_g + a_b)$, we have

$$o_i = c_i - a_i + \bar{a}. \quad (8)$$

This operation is visualized in Figure 5 for two vectors \mathbf{c} and \mathbf{a} .

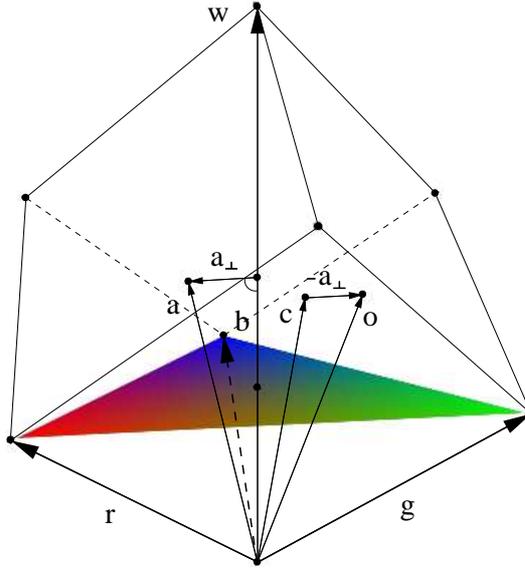


Fig. 5. First, the vector \mathbf{a} is projected onto the white vector \mathbf{w} . The projection is subtracted from \mathbf{a} which gives us \mathbf{a}_\perp , the component perpendicular to \mathbf{w} . This vector is subtracted from \mathbf{c} to obtain a color corrected image.

The entire algorithm can be realized easily in hardware. The averaging operation can be realized using a resistive grid [7,25]. See Koosh [37] for a realization

of analog computations in VLSI. We only require local connections. Therefore, the algorithm is scalable to arbitrary image sizes.

Instead of subtracting the component of space average color which is orthogonal to the white vector, one can also normalize both colors first

$$\hat{\mathbf{a}} = \frac{1}{a_r + a_g + a_b} [a_r, a_g, a_b]^T \quad (9)$$

$$\hat{\mathbf{c}} = \frac{1}{c_r + c_g + c_b} [c_r, c_g, c_b]^T \quad (10)$$

In this case, both space average color \mathbf{a} and the color of the current pixel \mathbf{c} are projected onto to the HSI plane $r + g + b = 1$ [35,38]. We again calculate the component of $\hat{\mathbf{a}}$ which is orthogonal to the vector \mathbf{w} .

$$\hat{\mathbf{a}}_{\perp} = \hat{\mathbf{a}} - (\hat{\mathbf{a}} \cdot \mathbf{w})\mathbf{w} \quad (11)$$

This component is subtracted from the normalized color vector $\hat{\mathbf{c}}$

$$\hat{\mathbf{o}} = \hat{\mathbf{c}} - \hat{\mathbf{a}}_{\perp} \quad (12)$$

or

$$\hat{o}_i = \hat{c}_i - \hat{a}_i + \frac{1}{3}. \quad (13)$$

The normalized output is then scaled using the illuminance component of the original pixel color.

$$o_i = (c_r + c_g + c_b)\hat{o}_i \quad (14)$$

$$= c_i - (c_r + c_g + c_b)\left(\hat{a}_i - \frac{1}{3}\right) \quad (15)$$

$$= c_i - \frac{c_r + c_g + c_b}{a_r + a_g + a_b}\left(a_i - \frac{1}{3}(a_r + a_g + a_b)\right) \quad (16)$$

$$= c_i - \frac{\bar{c}}{\bar{a}}(a_i - \bar{a}) \quad (17)$$

The calculations needed for this algorithm are shown in Figure 6.

5 Results

Both algorithms were tested on a series of images where different lighting conditions were used. Results for both algorithms are shown for an office scene in Figure 7. The images in the top row show the input image. The images in the second row show local space average color which was computed for each input image. The images in the third row show the output images of the first algorithm. In this case, the component of local space average color which is orthogonal to the gray vector was subtracted from the color of the input pixel. The images in the last row show the output images of the second algorithm. In this case,

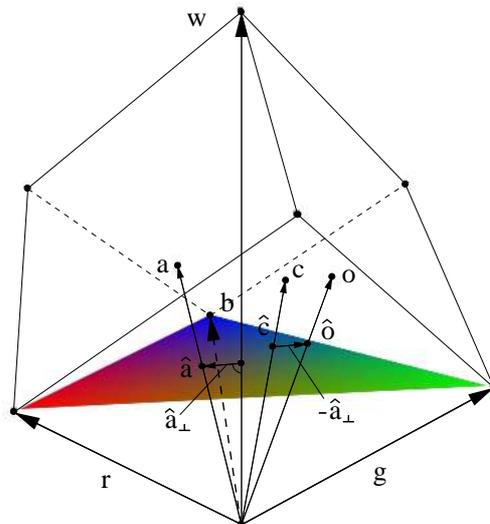


Fig. 6. First, the vectors \mathbf{c} and \mathbf{a} are projected onto the plane $r + g + b = 1$. The corresponding points are $\hat{\mathbf{c}}$ and $\hat{\mathbf{a}}$ respectively. Next, the vector $\hat{\mathbf{a}}$ is projected onto the white vector \mathbf{w} . The projection is subtracted from $\hat{\mathbf{a}}$ which gives us $\hat{\mathbf{a}}_{\perp}$, the component perpendicular to \mathbf{w} . This vector is subtracted from $\hat{\mathbf{c}}$ and finally scaled back to the original intensity to obtain a color corrected image.

local space average color and the color of the input pixel was normalized before performing the color correction step.

The input images were taken with a standard SLR camera. A CD-ROM with the images was produced when the film was developed. All three images show a desk with some utensils. The first image is very blue due to blue curtains which were closed at the time the image was taken. Sun was shining through the curtains which produced the blue background illumination. For the second input image a yellow light bulb was used to illuminate the room. Finally, the desk lamp was switched on for the third image. Again, the blue background illumination is caused by sunlight shining through the blue curtains. Note that the output images are much closer to what a human observed would expect.

6 Discussion

In the following we discuss differences to other color constancy algorithms which are related to the algorithm described in this contribution. The algorithms of Horn [6,7], Land [2], Moore et al. [25] and Rahman et al. [8] do not accurately reproduce human color perception. Helson [39] performed an extensive study with human subjects. The subjects task was to name the perceived color of gray stimuli which were illuminated with colored light. The stimuli were placed on a

Input Images:



Local Space Average Color:



Results for Algorithm 1:



Results for Algorithm 2:



Fig. 7. Experimental results for an office scene. The first row of images show the input images. The second row of images show local space average color. The third row shows the output images which were computed by subtracting the component of local space average color which is orthogonal to the gray vector from the color of the input pixel. The last row shows the results when local space average color and the color of the input pixel were normalized before performing the color correction step.

gray background. Helsons drew the following conclusions from his experiments. If the stimuli has a higher reflectance than the background, then the stimuli seems to have the color of the illuminant. If the stimuli has the same reflectance as the background then the stimuli is achromatic. If the stimuli has a lower

reflectance than the background then the subject perceives the stimuli as having the complementary color of the illuminant. The algorithms of Horn [6,7], Land [2], Moore et al. [25], and Rahman et al. [8] do not show this behavior. If the fraction between the color of the pixel and local space average color is computed then the color of the illuminant falls out of the equation and the stimuli will always appear to be achromatic.

All of the above methods require a normalization step which brings the output to the range $[0, 1]$. This normalization step can either be performed independently for each color band or the normalization can be performed uniformly across all color bands. In any case, one needs to loop over all pixels of the image. The algorithm which is described in this contribution does not require such a normalization step. All of the computations are performed inside the color cube. Values outside this color cube are clipped to the border of the cube.

In our previous work [32,34] we already discussed in depth the computation of local space average color using a grid of processing elements. Previously, we divided the color of the input pixel by the local space average color. This is exactly the gray world assumption [15,16] applied locally. This algorithm does not show the behavior described by Helson [39]. In [33] we subtract local space average color from the color of the input pixel followed by a rescaling operation. This method also does not correspond to human color perception described by Helson.

The algorithms described in this contribution performs color correction inside the unit color cube. The color cube is viewed as an Euclidian space. The color of the pixel is shifted in a direction perpendicular to the gray vector. The extent of the color shift is computed using local space average color. The first of the two algorithms which are described in this contribution shows the same response as a human observer for similar stimuli as was used in Helson's experiments.

7 Conclusion

In light of the current transition away from analog cameras towards digital cameras it is now possible to post-process the digital images before development to achieve accurate reproduction of the scene viewed. Such post-processing can either be done by the CPU of the camera or by post-processing the images on external hardware before the images are printed. Accurate color reproduction is very important for automatic object recognition. However, one of the largest markets will probably be consumer photography.

We have developed an algorithm for color constancy. The method consists of two parts: (a) a parallel grid of processing elements which is used to compute local space average color and (b) a method to estimate the original colors of the viewed objects. Instead of rescaling the red, green, and blue intensities using the inverse of local space average color, we shift the color vector into the direction of the gray vector. The color shift is based on local space average color. Therefore, the algorithm can also be used in the presence of varying illumination.

References

1. Land, E.H.: The retinex theory of colour vision. *Proc. Royal Inst. Great Britain* **47** (1974) 23–58
2. Land, E.H.: An alternative technique for the computation of the designator in the retinex theory of color vision. *Proc. Natl. Acad. Sci. USA* **83** (1986) 3078–3080
3. Brainard, D.H., Wandell, B.A.: Analysis of the retinex theory of color vision. In Healey, G.E., Shafer, S.A., Wolff, L.B., eds.: *Color*, Boston, Jones and Bartlett Publishers (1992) 208–218
4. Brill, M., West, G.: Contributions to the theory of invariance of color under the condition of varying illumination. *Journal of Math. Biology* **11** (1981) 337–350
5. Funt, B.V., Drew, M.S.: Color constancy computation in near-mondrian scenes using a finite dimensional linear model. In Jain, R., Davis, L., eds.: *Proceedings of the Computer Society Conference on Computer Vision and Pattern Recognition*, Ann Arbor, MI, Computer Society Press (1988) 544–549
6. Horn, B.K.P.: Determining lightness from an image. *Computer Graphics and Image Processing* **3** (1974) 277–299
7. Horn, B.K.P.: *Robot Vision*. The MIT Press, Cambridge, Massachusetts (1986)
8. Rahman, Z., Jobson, D.J., Woodell, G.A.: Method of improving a digital image. United States Patent No. 5,991,456 (1999)
9. Barnard, K., Finlayson, G., Funt, B.: Color constancy for scenes with varying illumination. *Computer Vision and Image Understanding* **65** (1997) 311–321
10. Forsyth, D.A.: A novel approach to colour constancy. In: *Second International Conference on Computer Vision* (Tampa, FL, Dec. 5-8), IEEE Press (1988) 9–18
11. Forsyth, D.A.: A novel algorithm for color constancy. In Healey, G.E., Shafer, S.A., Wolff, L.B., eds.: *Color*, Boston, Jones and Bartlett Publishers (1992) 241–271
12. Finlayson, G.D.: Color in perspective. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **18** (1996) 1034–1038
13. Barnard, K., Martin, L., Funt, B.: Colour by correlation in a three dimensional colour space. In Vernon, D., ed.: *Proceedings of the 6th European Conference on Computer Vision*, Dublin, Ireland, Berlin, Springer-Verlag (2000) 375–389
14. Finlayson, G.D., Hubel, P.M., Hordley, S.: Color by correlation. In: *Proceedings of IS&T/SID. The Fifth Color Imaging Conference: Color Science, Systems, and Applications*, Nov 17-20, The Radisson Resort, Scottsdale, AZ. (1997) 6–11
15. Buchsbaum, G.: A spatial processor model for object colour perception. *Journal of the Franklin Institute* **310** (1980) 337–350
16. Gershon, R., Jepson, A.D., Tsotsos, J.K.: From [R,G,B] to surface reflectance: Computing color constant descriptors in images. In McDermott, J.P., ed.: *Proc. of the 10th Int. Joint Conf. on Artificial Intelligence*, Milan, Italy. Volume 2., Morgan Kaufmann (1987) 755–758
17. Funt, B.V., Drew, M.S., Ho, J.: Color constancy from mutual reflection. *International Journal of Computer Vision* **6** (1991) 5–24
18. Ho, J., Funt, B.V., Drew, M.S.: Separating a color signal into illumination and surface reflectance components: Theory and applications. In Healey, G.E., Shafer, S.A., Wolff, L.B., eds.: *Color*, Boston, Jones and Bartlett Publishers (1992) 272–283
19. Maloney, L.T., Wandell, B.A.: Color constancy: a method for recovering surface spectral reflectance. *Journal of the Optical Society of America A3* **3** (1986) 29–33
20. D’Zmura, M., Lennie, P.: Mechanisms of color constancy. In Healey, G.E., Shafer, S.A., Wolff, L.B., eds.: *Color*, Boston, Jones and Bartlett Publishers (1992) 224–234

21. Courtney, S.M., Finkel, L.H., Buchsbaum, G.: A multistage neural network for color constancy and color induction. *IEEE Trans. on Neural Networks* **6** (1995) 972–985
22. Dufort, P.A., Lumsden, C.J.: Color categorization and color constancy in a neural network model of v4. *Biological Cybernetics* **65** (1991) 293–303
23. Funt, B., Cardei, V., Barnard, K.: Learning color constancy. In: *Proceedings of the IS&T/SID Fourth Color Imaging Conference, Scottsdale* (1996) 58–60
24. Herault, J.: A model of colour processing in the retina of vertebrates: From photoreceptors to colour opposition and colour constancy phenomena. *Neurocomputing* **12** (1996) 113–129
25. Moore, A., Allman, J., Goodman, R.M.: A real-time neural system for color constancy. *IEEE Transactions on Neural Networks* **2** (1991) 237–247
26. Novak, C.L., Shafer, S.A.: Supervised color constancy for machine vision. In Healey, G.E., Shafer, S.A., Wolff, L.B., eds.: *Color*, Boston, Jones and Bartlett Publishers (1992) 284–299
27. Usui, S., Nakauchi, S.: A neurocomputational model for colour constancy. In Dickinson, C., Murray, I., Carden, D., eds.: *John Dalton's Colour Vision Legacy. Selected Proc. of the Int. Conf., London*, Taylor & Francis (1997) 475–482
28. Finlayson, G.D., Schiele, B., Crowley, J.L.: Comprehensive colour image normalization. In Burkhardt, H., Neumann, B., eds.: *Fifth European Conf. on Computer Vision (ECCV '98)*, Freiburg, Germany, Berlin, Springer-Verlag (1998) 475–490
29. Cardei, V.C., Funt, B.: Committee-based color constancy. In: *Proceedings of the IS&T/SID Seventh Color Imaging Conference: Color Science, Systems and Applications*, Scottsdale, Arizona. (1999) 311–313
30. Ebner, M.: Evolving color constancy for an artificial retina. In Miller, J., Tomassini, M., Lanzi, P.L., Ryan, C., Tettamanzi, A.G.B., Langdon, W.B., eds.: *Genetic Programming: Proceedings of the 4th European Conference, Lake Como, Italy*, Berlin, Springer-Verlag (2001) 11–22
31. Risson, V.J.: Determination of an illuminant of digital color image by segmentation and filtering. United States Patent Application, Pub. No. US 2003/0095704 A1 (2003)
32. Ebner, M.: A parallel algorithm for color constancy. Technical Report 296, Universität Würzburg, Lehrstuhl für Informatik II, Am Hubland, 97074 Würzburg, Germany (2002)
33. Ebner, M.: Combining white-patch retinex and the gray world assumption to achieve color constancy for multiple illuminants. In Michaelis, B., Krell, G., eds.: *Pattern Recognition, Proceedings of the 25th DAGM Symposium*, Magdeburg, Germany, Berlin, Springer-Verlag (2003) 60–67
34. Ebner, M.: A parallel algorithm for color constancy. *Journal of Parallel and Distributed Computing* **64** (2004) 79–88
35. Gonzalez, R.C., Woods, R.E.: *Digital Image Processing*. Addison-Wesley Publishing Company, Reading, Massachusetts (1992)
36. Hanbury, A., Serra, J.: Colour image analysis in 3d-polar coordinates. In Michaelis, B., Krell, G., eds.: *Pattern Recognition. Proc. of the 25th DAGM Symposium*, Magdeburg, Germany, September 10-12, Berlin, Springer-Verlag (2003) 124–131
37. Koosh, V.F.: *Analog Computation and Learning in VLSI*. PhD thesis, California Institute of Technology Pasadena, California (2001)
38. Jain, R., Kasturi, R., Schunck, B.G.: *Machine Vision*. McGraw-Hill, Inc., New York (1995)
39. Helson, H.: Fundamental problems in color vision. i. the principle governing changes in hue, saturation, and lightness of non-selective samples in chromatic illumination. *Journal of Experimental Psychology* **23** (1938) 439–476