Combining White-Patch Retinex and the Gray World Assumption to Achieve Color Constancy for Multiple Illuminants

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 irrespective of the actual light they reflect. This ability to compute color constant descriptors is an important problem for computer vision research. We have developed a parallel algorithm for color constancy. The algorithm is based on two fundamental theories of color constancy, the gray world assumption and the white-patch retinex algorithm. The algorithm's performance is demonstrated on several images where objects are illuminated by multiple illuminants.

1 Motivation

The human visual system is able to correctly determine the color of objects irrespective of the actual light reflected by the objects. For instance, if a white wall is illuminated with red light, it will reflect more red light in comparison to the amount of light reflected in the green and blue spectrum. If the same wall is illuminated with green light, then the wall will reflect more light towards the green spectrum. If the scene viewed by a human observer is sufficiently complex, the wall will nevertheless appear white to a human observer. The human visual system is somehow able to discount the illuminant and to estimate the reflectances of the objects in view [24]. This ability is called color constancy, as the perceived color remains constant irrespective of the illuminant. Two different mechanisms may be used by the human visual system to achieve color constancy [20]. We have devised a parallel algorithm which is based on both of these mechanism. Previously, we have only used the gray world assumption [8].

Numerous solutions to the problem of color constancy have been proposed. Land, a pioneer in color constancy research has proposed the retinex theory [19]. Others have added to this research and proposed variants of the retinex theory [2, 3, 16, 18]. Other algorithms for color constancy include gamut-constraint methods [1, 13], perspective color constancy [10], color by correlation [11], the gray world assumption [4, 17], recovery of basis function coefficients [21], mechanisms of light adaptation coupled with eye movements [7], neural networks [6, 15, 22], minimization of an energy function [23], comprehensive color normalization [12], committee-based methods which combine the output of several different color constancy algorithms [5] or use of genetic programming [9]. Most solutions to color constancy only assume a single illuminant. Our algorithm can also cope with multiple illuminants. It runs on a parallel grid of simple processing elements which only perform local computations. No global computations are made. Thus, it is scalable and lends itself to a VLSI implementation.

2 Color Image Formation

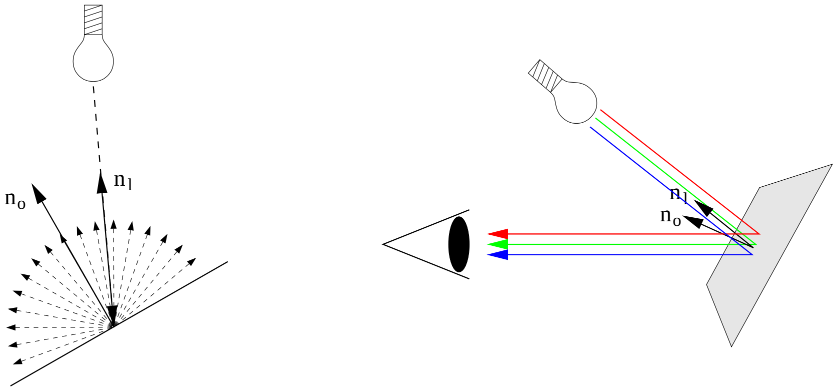

Fig. 1. For a Lambertian surface the amount of reflected light does not depend on viewing angle (left). It only depends on the angle between the surface normal and the direction of the light source. Part of the incoming light is absorbed by the surface, the remainder is reflected equally in all directions. We assume that the sensor's response function is described by a delta function. Thus only three different wavelength (red, green and blue) need to be considered (right).

Assume that we have an object with a Lambertian surface. Let a ray of light with intensity $L(\lambda)$ and wavelength λ be reflected by this object. Let $\mathbf{x}_o$ be the position where the ray hits the object. Part of the light is absorbed by the object, the remainder is reflected equally in all direction. For a Lambertian surface the light reaching the eye does not depend on viewing angle. It only depends on the angle between the surface normal and the direction of the light source (Figure 1). The response of a sensor at position $\mathbf{x}_s$ which measures the reflected ray is given by

$$\mathbf{I}(\mathbf{x}_s) = \mathbf{n}_l \cdot \mathbf{n}_o \int_\lambda R(\lambda, \mathbf{x}_o) L(\lambda) \mathbf{S}(\lambda) d\lambda \qquad (1)$$

where $\mathbf{I}(\mathbf{x}_s)$ is a vector of sensor responses, $\mathbf{n}_l$ is the unit vector pointing in the direction of the light source, $\mathbf{n}_o$ is the unit vector corresponding to the surface normal, $R(\lambda, \mathbf{x}_o)$ specifies the percentage of light reflected by the surface, and $\mathbf{S}(\lambda)$ specifies the sensor's response functions [12]. The sensor's response is calculated by integrating over all wavelengths to which the sensor responds.

If we assume ideal sensors for red, green and blue light, the sensor's response function is given by a delta function $(S_i(\lambda) = \delta(\lambda - \lambda_i))$ with $i \in \{\text{red}, \text{green}, \text{blue}\}$. If we also assume that the light source illuminates the surface at a right angle, the above equation simplifies to

$$I_i(\mathbf{x}_s) = R(\lambda_i, \mathbf{x}_o)L(\lambda_i) \tag{2}$$

where $I_i(\mathbf{x}_s)$ denotes the i-th component of the vector $\mathbf{I}(\mathbf{x}_s)$. Thus, the light which illuminates the scene is scaled by the reflectances.

The light illuminating the scene can be recovered easily if the image contains at least one pixel for each band which reflects all light for this particular band. We only need to loop over all pixel values, and record the maximum intensity values for all three bands. Using these three values we rescale all color bands to the range $[0, 1]$.

$$R(\lambda_i, \mathbf{x}_o) = \frac{I_i(\mathbf{x}_s)}{L_{\max}(\lambda_i)} \tag{3}$$

with $L_{\max}(\lambda_i) = \max_{\mathbf{x}}\{I_i(\mathbf{x})\}$. This algorithm is called the white-patch retinex algorithm [14].

A second algorithm for color constancy is based on the assumption that the average color is gray. If we assume that the reflectances of the surface are uniformly distributed over the interval $[0, 1]$, the average value will be 0.5 for all bands [9].

$$\begin{aligned} \frac{1}{N}\sum_{\mathbf{x}}^{N} I_i(\mathbf{x}) &= \frac{1}{N}\sum_{\mathbf{x}}^{N} R(\lambda_i, \mathbf{x})L(\lambda_i) \\ &= L(\lambda_i)\frac{1}{N}\sum_{\mathbf{x}}^{N} R(\lambda_i, \mathbf{x}) \\ &= L(\lambda_i)\frac{1}{2} \end{aligned} \tag{4}$$

Thus, space average color can be used to estimate the intensities of the light illuminating the scene. The light illuminating the scene is simply twice the space average color.

$$L(\lambda_i) = \frac{2}{N}\sum_{\mathbf{x}}^{N} I_i(\mathbf{x}) \tag{5}$$

The reflectances can then be calculated as follows.

$$R(\lambda_i, \mathbf{x}_o) = \frac{I_i(\mathbf{x}_s)}{L(\lambda_i)} \tag{6}$$

Both cues, space-average scene color as well as the color of the highest luminance patch may be used by the human visual system to estimate the color of the light illuminating the scene [20].

3 Calculating Local Space Average Color

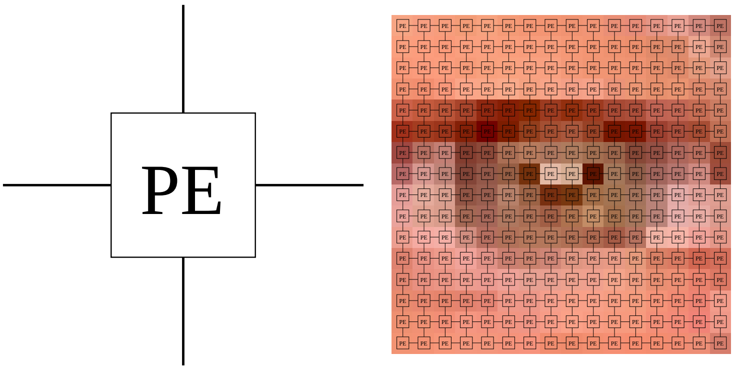

Fig. 2. Each processing element is connected to four neighbors (left). We have one processing element for each pixel of the input image (right).

Our algorithm runs on a parallel grid of processing elements. Each processing element is connected to four other processing elements (Figure 2). We have one element per pixel. A single element is connected to the elements on the left, on the right as well as to the elements above and below the current element. For each color band red, green, and blue, we calculate local space average color by averaging data from the four neighboring elements and slowly fading the intensity of the current band into the result. Let $\mathbf{c}(x,y) = [c_{\text{red}}(x,y), c_{\text{green}}(x,y), c_{\text{blue}}(x,y)]$ be the color of the pixel at position (x,y) and $\mathbf{avg}(x,y) = [\text{avg}_{\text{red}}(x,y), \text{avg}_{\text{green}}(x,y), \text{avg}_{\text{blue}}(x,y)]$ be local space average color estimated by element (x,y). Let p_1 be a small percentage. Local space average color is computed by iterating the following equation indefinitely for all three bands $i \in \{\text{red}, \text{green}, \text{blue}\}$.

$$\begin{aligned} a_i(x,y) &= \frac{1}{4}(\text{avg}_i(x-1,y) + \text{avg}_i(x,y-1) + \text{avg}_i(x+1,y) + \text{avg}_i(x,y+1)) \\ \text{avg}_i(x,y) &= (1-p_1)a_i(x,y) + p_1 \cdot c_i(x,y) \end{aligned} \tag{7}$$

In case of a static image, we can stop the calculations after the difference between the old and the new estimate has been reduced to a small value. A sample calculation for a scene illuminated with two different illuminants is shown in Figure 3.

The calculations are done independently for all three color bands red, green, and blue. The first term averages the data from neighboring elements and multiplies the result with $(1-p_1)$. The second term is the local color multiplied by a small percentage p_1. This operation slowly fades the local color into the current estimate of the local space average color. The factor p_1 determines the extent over which local space average color will be computed. As local average color is

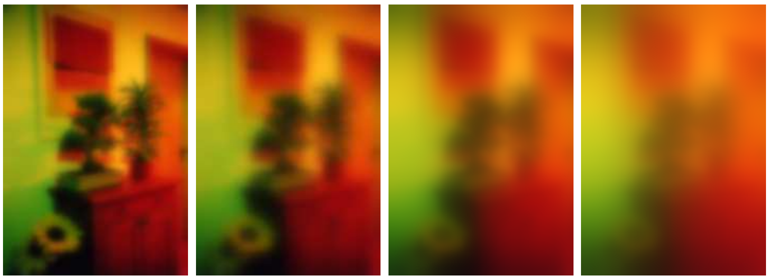

Fig. 3. Space average color after 50, 200, 1000 and 5000 iterations of the algorithm.

handed from one element to the next, it will be multiplied by $(1-p_1)$. Thus, if p_1 is large, the influence of local space average color will decay very fast from one element to the next. On the other hand, if p_1 is small, then it will decay very slowly.

4 Parallel Dynamic Range Estimation

We now have local average color and the input color available at every processing element. In order to restore the original colors of the image we look at the deviation from local average color. Let $d_i(x,y)$ be the deviation between local average color and the current color at position (x,y). We take the maximum across neighboring elements:

$$d_i'(x,y) = \max\{|\text{avg}_i - c_i|, d_i(x-1,y), d_i(x,y-1), d_i(x+1,y), d_i(x,y+1)\} \quad (8)$$

Finally, we reduce the maximum deviation by a small percentage p_2.

$$d_i = (1-p_2)d_i' \quad (9)$$

The factor p_2 determines how fast the deviation decays to zero as it is passed from element to element. This deviation is used to scale the difference between the current color and local space average color.

$$o_i = \frac{(c_i - \text{avg}_i)}{d_i} \quad (10)$$

Finally a sigmoidal activation function is used to transform the computed value to the range [0,1].

$$r_i = \frac{1}{1+e^{-\frac{o_i}{\sigma}}} \quad (11)$$

If o_i is close to zero, i.e. local average color and the color of the current pixel is very similar, then the output color r_i will be gray. We also experimented with a linear transformation. In this case, output color is computed as follows:

$$r_i = \frac{1}{2}(1+o_i) \quad (12)$$

Values smaller than 0 are set to zero and values larger than 1 are set to 1. The difference between the sigmoidal and linear transformation are shown in Figure 4. Use of the sigmoidal transfer function produced better results.

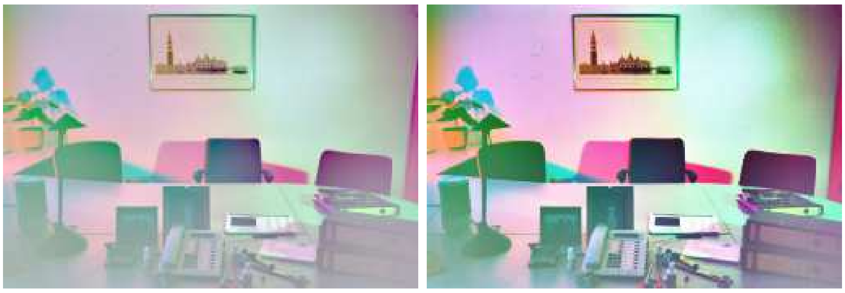

Fig. 4. A linear output function was used for the left image. A sigmoidal output function was used for the right image. The colors of the left image look less saturated than the colors of the right image.

5 Results

The above algorithm was tested on several real world images. In each case multiple illuminants were used to illuminate the objects in the scene. The images were taken with an analog SLR camera, developed on film and then digitized. The digitized images were linearized with a gamma correction of 2.2. The algorithm was run on the linearized input images. A gamma correction of $\frac{1}{2.2}$ was applied to the output images. The following parameters were used: $p_1 = 0.0005$, $p_2 = 0.005$, $\sigma = 0.2$. The size of the input images was 256×175 pixels. Results for four different input images are shown in Figure 5. The first row shows the input images. The second row shows local average color, the third row shows the absolute deviation from local average color and the last row shows the output images of our algorithm. The first three images show objects illuminated with two colored light bulbs. For the fourth image, the camera's built in flash was used to illuminate the scene. As can be seen from the output images, the algorithm is able to adjust the colors of the input images. For a scene illuminated with white light the output is almost equivalent to the input image.

6 Conclusion

We have developed a parallel algorithm for color constancy. The algorithm calculates local space average color and maximum deviation of the current color from local average color. Both cues are used to estimate the reflectances of the

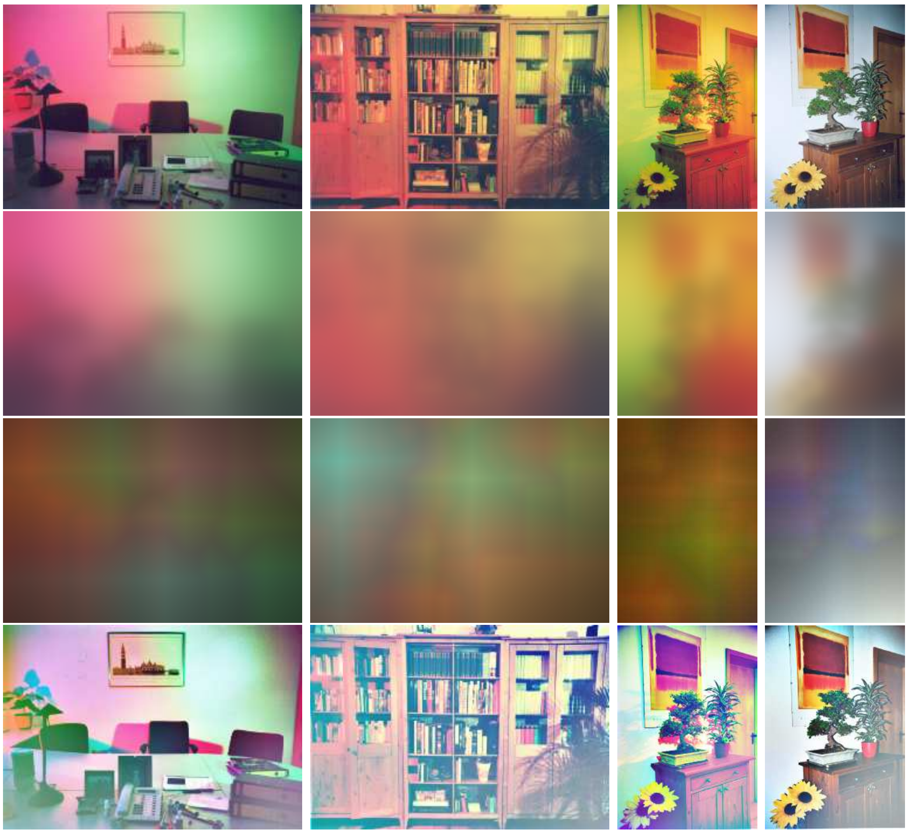

Fig. 5. Results for 4 different input images. Two colored illuminants were used for the first three images. A flash was used to illuminate the objects shown in the last image.

objects in view. In this respect, the algorithm is a combination of both the gray world assumption and the white patch retinex algorithm. The algorithm's ability to estimate the reflectances of the objects in view was demonstrated on several real world images taken with multiple illuminants.

References

1. K. Barnard, G. Finlayson, and B. Funt. Color constancy for scenes with varying illumination. *Computer Vision and Image Understanding*, 65(2):311–321, 1997.
2. D. H. Brainard and B. A. Wandell. Analysis of the retinex theory of color vision. In G. E. Healey, S. A. Shafer, and L. B. Wolff, eds., *Color*, pp. 208–218, Boston, 1992. Jones and Bartlett Publishers.
3. M. Brill and G. West. Contributions to the theory of invariance of color under the condition of varying illumination. *Journal of Math. Biology*, 11:337–350, 1981.

4. G. Buchsbaum. A spatial processor model for object colour perception. *Journal of the Franklin Institute*, 310(1):337–350, 1980.
5. V. C. Cardei and B. Funt. Committee-based color constancy. In *Proc. of the IS&T/SID 7th Color Imaging Conference: Color Science, Systems and Applications*, pp. 311–313, 1999.
6. S. M. Courtney, L. H. Finkel, and G. Buchsbaum. A multistage neural network for color constancy and color induction. *IEEE Trans. on Neural Networks*, 6(4):972–985, 1995.
7. M. D'Zmura and P. Lennie. Mechanisms of color constancy. In Glenn E. Healey, Steven A. Shafer, and Lawrence B. Wolff, eds., *Color*, pp. 224–234, Boston, 1992. Jones and Bartlett Publishers.
8. M. Ebner. A parallel algorithm for color constancy. Technical Report 296, Universität Würzburg, Lehrstuhl für Informatik II, Würzburg, Germany, April 2002.
9. M. Ebner. Evolving color constancy for an artificial retina. In J. Miller, M. Tomassini, P. Luca Lanzi, C. Ryan, A. G. B. Tettamanzi, and W. B. Langdon, eds., *Genetic Programming: Proc. of the 4th Europ. Conf., EuroGP 2001, Lake Como, Italy*, pp. 11–22, Berlin, 2001. Springer-Verlag.
10. G. D. Finlayson. Color in perspective. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 18(10):1034–1038, 1996.
11. G. D. Finlayson, P. M. Hubel, and S. Hordley. Color by correlation. In *Proc. of IS&T/SID. The 5th Color Imaging Conference: Color Science, Systems, and Applications, The Radisson Resort, Scottsdale, AZ*, pp. 6–11, 1997.
12. G. D. Finlayson, B. Schiele, and J. L. Crowley. Comprehensive colour image normalization. In *Fifth Europ. Conf. on Computer Vision*, 1998.
13. D. A. Forsyth. A novel approach to colour constancy. In *2nd Int. Conf. on Computer Vision, Tampa, FL*, pp. 9–18. IEEE Press, 1988.
14. B. Funt, K. Barnard, and L. Martin. Is colour constancy good enough? In *Fifth Europ. Conf. on Computer Vision*, pp. 445–459, 1998.
15. B. Funt, V. Cardei, and K. Barnard. Learning color constancy. In *Proc. of the IS&T/SID 4th Color Imaging Conference*, pp. 58–60, Scottsdale, 1996.
16. B. V. Funt and M. S. Drew. Color constancy computation in near-mondrian scenes using a finite dimensional linear model. In *Proc. of the Comp. Society Conf. on Computer Vision and Pattern Recognition*, pp. 544–549. Comp. Society Press, 1988.
17. R. Gershon, A. D. Jepson, and J. K. Tsotsos. From [r,g,b] to surface reflectance: Computing color constant descriptors in images. In *Proc. of the 10th Int. Joint Conference on Artificial Intelligence*, volume 2, pp. 755–758, 1987.
18. B. K. P. Horn. *Robot Vision.* The MIT Press, Cambridge, Massachusetts, 1986.
19. E. H. Land. The retinex theory of colour vision. *Proc. Royal Inst. Great Britain*, 47:23–58, 1974.
20. K. J. Linnell and D. H. Foster. Space-average scene colour used to extract illuminant information. In C. Dickinson, I. Murray, and D. Carden, eds., *John Dalton's Colour Vision Legacy. Selected Proc. of the Int. Conf.*, pp. 501–509, London, 1997. Taylor & Francis.
21. L. T. Maloney and B. A. Wandell. Color constancy: a method for recovering surface spectral reflectance. *Journal of the Opt. Society of America A3*, 3(1):29–33, 1986.
22. A. Moore, J. Allman, and R. M. Goodman. A real-time neural system for color constancy. *IEEE Trans. on Neural Networks*, 2(2):237–247, 1991.
23. S. Usui and S. Nakauchi. A neurocomputational model for colour constancy. In C. Dickinson, I. Murray, and D. Carden, eds., *John Dalton's Colour Vision Legacy. Selected Proc. of the Int. Conf.*, pp. 475–482, London, 1997. Taylor & Francis.
24. S. Zeki. *A Vision of the Brain.* Blackwell Science, Oxford, 1993.