Estimating the Color of the Illuminant using Anisotropic Diffusion

Marc Ebner

Universität Würzburg, Lehrstuhl für Informatik II
Am Hubland, 97074 Würzburg, Germany
ebner@informatik.uni-wuerzburg.de
http://wwwi2.informatik.uni-wuerzburg.de/staff/ebner/welcome.html

Abstract. The human visual system is able to determine the color of objects irrespective of the power distribution illuminating the scene. This ability is called color constancy. It would be highly interesting to mimic this ability of the human visual system. Accurate color reproduction is very important from consumer photography to automatic color based object recognition. In theory, if we knew the color of the illuminant for each image pixel, we would be able to compute a color corrected image which is independent of the illuminant. We suggest the use of anisotropic diffusion to estimate the illuminant locally for each image pixel.

1 Motivation

The human visual system is able to determine the color of objects irrespective of the type of illuminant used. The ability to compute color constant descriptors from the light entering the eye is called color constancy [1,2]. In contrast, colors measured by a digital sensor are not constant. The response of the sensor depends on the color of the illuminant. Obtaining accurate colors of objects is very important from consumer photography to automatic object recognition based on color. In general, accurate color rendition is required for many algorithms in color image processing.

In most cases, we have multiple illuminants. For instance, we may have artificial lighting inside a room and additionally daylight falling through a window. If we have several different light sources, then the illuminant is no longer uniform over the entire image. In this case, we have to estimate the illuminant locally for each image pixel [3,4]. So far, only a few algorithms have been developed which address the problem of a locally varying illuminant. Most of the color constancy literature deals with a uniform illuminant. The algorithms which also take a spatially varying illuminant into account are the algorithms of Land and McCann [5], Horn [6], Blake [7], Moore et al. [8], Rahman et al. [9], Barnard et al. [10], and Ebner [4].

2 Locally Estimating the Color of the Illuminant

Ebner [4] has developed a parallel algorithm for color constancy which estimates the illuminant locally for each image pixel. The algorithm runs on a grid of processing elements and uses isotropic diffusion to compute local space average color. Let $\mathbf{a}$ be the current estimate of local space average color and let $\mathbf{c}$ be the color of the pixel stored at the current processing element. The data which is available from neighboring elements is averaged and then a little amount from the color of the current element is added to the average. Let $N(x,y)$ be the neighboring processing elements of element (x,y). The following two update equations are iterated until convergence

$$\mathbf{a}'(x,y) = \frac{1}{|N(x,y)|} \sum_{(x',y') \in N(x,y)} \mathbf{a}(x',y') \tag{1}$$

$$\mathbf{a}(x,y) = \mathbf{a}'(x,y) \cdot (1-p) + \mathbf{c}(x,y) \cdot p \tag{2}$$

where p is a small percentage larger than zero. During each iteration a new estimate of local space average color is computed. The parameter p determines the extent over which local space average color is computed. For small values of p, local space average color is computed over a large area. For large values of p, local space average color is computed for a small area. For the gray world assumption to hold, we need to compute local space average color for a comparatively large area.

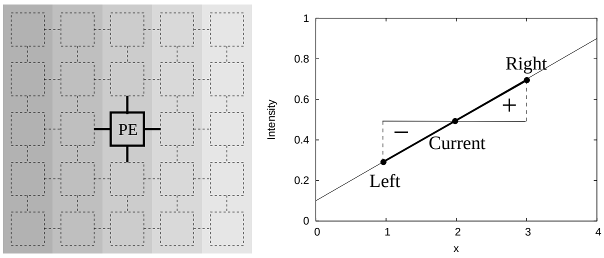

Fig. 1. Let us assume a linearly changing illuminant and consider the processing element located at the center of the grid. The differences between the averages computed by this processing element will cancel which results in a series of vertical bands.

Let us now assume that the illuminant changes smoothly over the entire image from left to right. Figure 1 illustrates such a smooth horizontal change of the illuminant from left to right. Consider the data computed by the processing element in the center of the image. The average computed by the element on the left will be slightly lower whereas the average computed by the element on the right will be slightly higher than the current average. The data computed by processing elements above and below will be equivalent to the data computed by the current element. If we average the data from the neighboring elements, the differences between the element on the left and the element on the right will cancel. After convergence, one obtains a series of vertical bands. In other words, the estimate of local space average color will be correct, provided that we have a linearly changing illuminant.

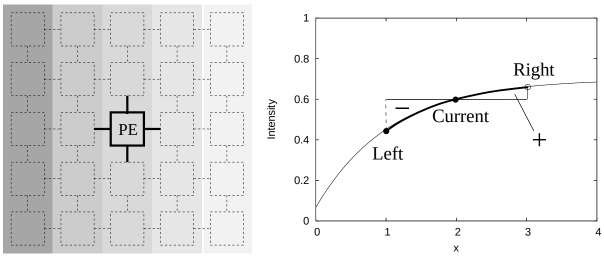

Fig. 2. Assuming that we have a non-linear change of the illuminant from left to right, then the average computed by the processing element to the left and the average computed by the processing element on the right will not be equivalent. The estimate computed by the processing element located in the center will be too low.

Let us now assume that we have a non-linear change of the illuminant as shown in Figure 2. In this case, the average computed by the element on the left and on the right will no longer cancel. The estimate of the illuminant will be slightly too low. We can solve this problem by computing space average color only along the vertical. Thus, we would be estimating the color of the illuminant independently for each column of the image. We could also try to reduce the area of support. For our parallel algorithm, the area of support is determined by the parameter p. If we make the area of support sufficiently small, then the assumption that the illuminant varies linearly may be correct. In practice, it may not be possible to make the area of support very small because then the gray world assumption may no longer hold. However, if the area of support is very large, then the area of support will likely straddle a non-linear transition of the illuminant.

3 Use of Anisotropic Diffusion

A more accurate estimate of the illuminant can be obtained if the averaging is done non-uniformly. This leads us to anisotropic diffusion which is frequently used to segment noisy images [11,12,13]. For our example, we have assumed a horizontally varying illuminant. The illuminant is constant along the vertical. Let us call these vertical lines the lines of constant illumination or iso-illumination lines. If the illuminant were changing from top to bottom, the iso-illumination lines would be oriented horizontally. In this case, we could average the data along the lines of the image. In general, of course, the change of the illuminant may be arbitrary across the image. A spotlight may cause a circular or elliptic change of the illuminant.

Suppose that we knew how the iso-illumination lines run across the image. Then we could establish an estimate of the illuminant by averaging the data only along the iso-illumination lines. In other words, we suggest the use of anisotropic diffusion in order to estimate the illuminant locally for each image pixel. If we only average the data in a direction which is perpendicular to the illumination

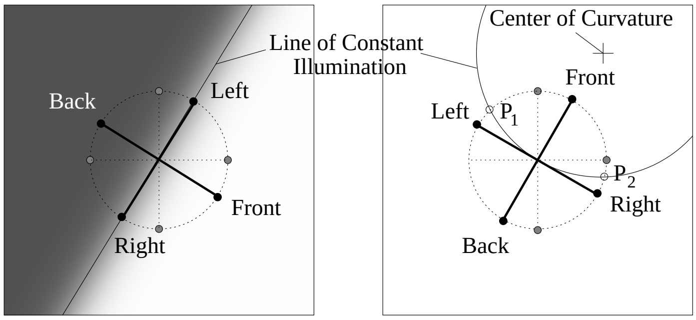

Fig. 3. Rotated coordinate system where the directions left/right point along the line of constant illumination. The directions front/back point along the gradient. A curved line of constant illumination is shown on the right.

gradient then the result will be independent of the area of support provided that it is sufficiently large. This is a very important point. If we average the data along the iso-illumination lines, the scaling parameter p could be made arbitrarily small. It would only depend on the size of the image. In contrast, if we compute isotropic local space average color, the area of support should be small for areas where the illuminant is changing very much and it would have to be large for areas where the illuminant is constant. If we compute isotropic local space average color along the iso-illumination lines then the estimate of the illuminant will be correct also for a non-linearly changing illuminant.

Let us define a local coordinate system for each image pixel as shown in Figure 3 on the left. For each pixel we have four directions. Two are perpendicular to the iso-illumination line and two point along the iso-illumination line. The gradient of the illuminant defines this coordinate system for each image pixel. The problem is that we need to have some estimate of the illuminant in order to compute the gradient. We can use isotropic diffusion to get a rough estimate of the illuminant. This estimate can then be used to compute the gradient. Let $\mathbf{a}$ be the estimate computed using isotropic diffusion or some other method such as applying a Gaussian filter to the input image. Let $[dx_i, dy_i]$ be the gradient of color channel i.

$$\nabla a_i = \begin{bmatrix} dx_i \\ dy_i \end{bmatrix} = \begin{bmatrix} \frac{\partial a_i}{\partial x} \\ \frac{\partial a_i}{\partial y} \end{bmatrix} . \tag{3}$$

Since we have a three-band color image, we would obtain slightly different gradients for each color band. We assume that a single gradient $[dx, dy]$ describes the actual distribution of the illuminant. Thus, we need to combine the three gradients into one. One way to do this is to simply average the gradients.

$$\begin{bmatrix} dx \\ dy \end{bmatrix} = \frac{1}{3} \sum_{i \in \{r,g,b\}} \nabla a_i \tag{4}$$

It would also be possible to use different weights for the three channels. We could also apply weights based on the magnitude of the gradient.

Let $\mathbf{a}$(front), $\mathbf{a}$(back), $\mathbf{a}$(left), and $\mathbf{a}$(right) be the estimate of space average color in the corresponding directions front, back, left and right as shown in Figure 3. Let $\mathbf{c}(x, y)$ be the color at the current element. It is assumed that the locations front, back, left and right are a unit distance away from the current element. The data at these locations is computed using bilinear interpolation. We can calculate local space average color $\mathbf{a}$ by averaging the existing estimates obtained from the left and the right along the line of constant illumination

$$\mathbf{a}'(x, y) = \frac{1}{3}(\mathbf{a}(x, y) + \mathbf{a}(\text{left}) + \mathbf{a}(\text{right})) \tag{5}$$

$$\mathbf{a}(x, y) = \mathbf{a}'(x, y) \cdot (1 - p) + \mathbf{c}_i(x, y) \cdot p \tag{6}$$

where p is a small percentage.

Data from the front/back direction can also be included by introducing an additional parameter q with $q \in [0, \frac{1}{6}]$ which describes the exchange along the front/back direction.

$$\mathbf{a}'(x, y) = \left(\frac{1}{3} - q\right) \mathbf{a}(\text{left}) + \left(\frac{1}{3} - q\right) \mathbf{a}(\text{right}) + \frac{1}{3}\mathbf{a}(x, y) + q\mathbf{a}(\text{front}) + q\mathbf{a}(\text{back}) \tag{7}$$

If q is equal to zero, then we average data only along the line of constant illumination. If q is larger than zero, some data will also flow along the front/back direction. The reader should be aware that the update equations are given for the center elements only. Processing elements which are located on the boundary of the image can only average the data which is available.

Note that the line of constant illumination may not be a straight line. In case of a spotlight it may be circular or elliptic. For actual images which contain local light sources this is likely to be the case. Thus, we have to determine the curvature for each image pixel. The curvature K of a point (x, y) on a surface $F(x, y)$ is defined as [14]

$$K = \frac{\begin{vmatrix} F_{xx} & F_{xy} & F_x \\ F_{yx} & F_{yy} & F_y \\ F_x & F_y & 0 \end{vmatrix}}{(F_x^2 + F_y^2)^{3/2}} \tag{8}$$

with $F_x = \frac{\partial F}{\partial x}$, $F_y = \frac{\partial F}{\partial y}$, $F_x = \frac{\partial F}{\partial x}$, $F_{xy} = \frac{\partial F}{\partial x \partial y}$, $F_{yx} = \frac{\partial F}{\partial y \partial x}$, $F_{xx} = \frac{\partial^2 F}{\partial x^2}$, $F_{yy} = \frac{\partial^2 F}{\partial y^2}$. Let dx and dy be the gradient computed from the estimated illuminant using isotropic diffusion. The curvature of the illuminant is then computed by setting $F_x = dx$, $F_y = dy$, $F_{xy} = \frac{\partial}{\partial x}dy$, $F_{yx} = \frac{\partial}{\partial y}dx$, $F_{xx} = \frac{\partial}{\partial x}dx$ and $F_{yy} = \frac{\partial}{\partial y}dy$. From the curvature K, we can calculate the radius r of the curve [14]

$$r = \left|\frac{1}{K}\right| = \left|\frac{(F_x^2 + F_y^2)^{3/2}}{F_x F_{xy} F_y + F_x F_{yx} F_y - F_x^2 F_{yy} - F_y^2 F_{xx}}\right|. \tag{9}$$

The center of the curvature lies on the positive side of the curve normal if $K > 0$ and on the negative side otherwise. If $K = 0$ then the line of constant illumination is indeed a straight line.

The two points along the line of constant illumination from which we need to extract our current estimate of local space average color can be found as shown in Figure 3 on the right. We need to determine the intersection points between the unit circle and the circle determined from the current estimate of the illuminant. This will give us the two values $\mathbf{a}(P_1)$ and $\mathbf{a}(P_2)$ which can be used in the anisotropic averaging operation as described above. In order to compute the intersection points we assume that the center of the curvature is located at point $(0, r)$. Then it is straight forward to compute the location of the two intersection points P_1 and P_2 from the two circles which are given by

$$x^2 + y^2 = 1 \qquad \text{and} \qquad (x - r)^2 + y^2 = r^2. \tag{10}$$

We solve for x and obtain $x = \frac{1}{2r}$. The y coordinates are obtained from the equation of the unit circle. We obtain $y_{1/2} = \pm\sqrt{1 - \frac{1}{4r^2}}$. Of course, in the general case, the center of the curvature does not lie on the X axis. In this case, we simply perform an appropriate rotation of the coordinate system. Local space average color at points P_1 and P_2 are again obtained using bilinear interpolation. Let $\check{a}_i$ be the previous estimate of local space average color at points P_1 and P_2. We compute

$$\check{\mathbf{a}}'(x, y) = \frac{1}{3}(\check{\mathbf{a}}(P_1) + \check{\mathbf{a}}(x, y) + \check{\mathbf{a}}(P_2)). \tag{11}$$

Once the data from neighboring elements is averaged, the color from the current element is slowly faded into the result

$$\check{\mathbf{a}}(x, y) = \check{\mathbf{a}}'(x, y) \cdot (1 - p) + \mathbf{c}(x, y) \cdot p. \tag{12}$$

Again, p is a small value larger than zero. These update equations are iterated for each processing element until convergence. We can now handle curved non-linear transitions of the illuminant by averaging the data along the line of constant illumination.

4 Experimental Results

Barnard et al. [15] created a set of images to be used for color constancy research. We took two images from the data set which show the same scene under two different illuminants and combined them artificially. The illuminant of the first image is bluish. The illuminant of the second image is basically white. Figure 4 shows the combined images. A horizontal gradient was used to create the first image whereas a spotlight effect was simulated for the second image. The intensity of the of the first image was increased by a factor of three. Apart from having a color gradient, we thus also have an intensity gradient.

Figure 4 shows the results for the algorithm described above. The parameter p was set to 0.0001. The parameter q was set to 0. The illuminant is roughly approximated by computing a blurred input image using exponential blur. The intensity was used to estimate the illuminant. For the first image, we obtain horizontal lines of constant illumination, whereas for the second image, we obtain

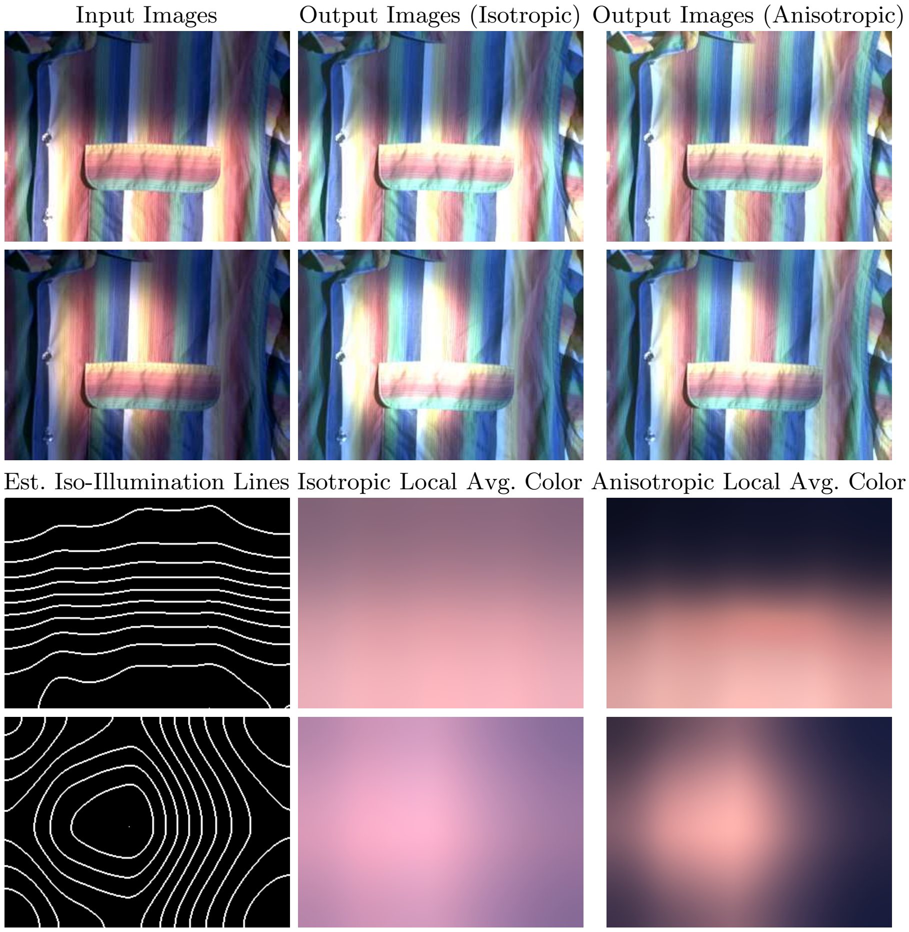

Fig. 4. Two images from the database of Barnard et al. [15] were merged to simulate a horizontal illumination gradient and a spotlight effect. Results are shown using isotropic diffusion and anisotropic diffusion.

circular lines of constant illumination. Next, we use anisotropic diffusion to estimate the illuminant locally for each image pixel. When we compare the estimate of the illuminant which was obtained using isotopic diffusion with the estimate of the illuminant which was obtained using anisotropic computation, we see that computation of local space average color using anisotropic diffusion more accurately estimates the color of the illuminant. Once the illuminant is known, we can compute a color corrected image. The resulting output images are shown in Figure 4. When comparing the two output images we see that some detail is missing from the resulting image when local space average color is computed using isotropic local space average color. In contrast, when local space average color is computed using anisotopic diffusion more detail is retained.

5 Conclusion

Many color constancy algorithms have been proposed. Few of the algorithms address the problem of estimating the illuminant locally for each image pixel. Most algorithms assume that the color of the illuminant is constant over the entire image. We have shown how anisotropic diffusion may be used to estimate the illuminant locally. First the illuminant is roughly approximated using Gaussian or exponential blur, then a refined estimate of the illuminant is computed. The algorithm runs on a grid of processing elements where data is exchanged only between neighboring elements. Each element estimates the line of constant illumination and then averages data along this line. As a result, we obtain an estimate which may also be used if the illuminant varies non-linearly across the image.

References

1. Zeki, S.: A Vision of the Brain. Blackwell Science, Oxford (1993)
2. Ebner, M.: Color Constancy. John Wiley & Sons, England (2007)
3. Ebner, M.: Color constancy using local color shifts. In Pajdla, T., Matas, J., eds.: Proceedings of the 8th European Conference on Computer Vision, Part III,Prague, Czech Republic, May, 2004, Berlin, Springer-Verlag (2004) 276–287
4. Ebner, M.: A parallel algorithm for color constancy. Journal of Parallel and Distributed Computing **64** (2004) 79–88
5. Land, E.H., McCann, J.J.: Lightness and retinex theory. Journal of the Optical Society of America **61** (1974) 1–11
6. Horn, B.K.P.: Determining lightness from an image. Computer Graphics and Image Processing **3** (1974) 277–299
7. Blake, A.: Boundary conditions for lightness computation in mondrian world. Computer Vision, Graphics, and Image Processing **32** (1985) 314–327
8. Moore, A., Allman, J., Goodman, R.M.: A real-time neural system for color constancy. IEEE Transactions on Neural Networks **2** (1991) 237–247
9. Rahman, Z., Jobson, D.J., Woodell, G.A.: Method of improving a digital image. United States Patent No. 5,991,456 (1999)
10. Barnard, K., Finlayson, G., Funt, B.: Color constancy for scenes with varying illumination. Computer Vision and Image Understanding **65** (1997) 311–321
11. Didas, S., Weickert, J., Burgeth, B.: Stability and local feature enhancement of higher order nonlinear diffusion filtering. In Wropatsch, W., Sablatnig, R., Hanbury, A., eds.: Pattern Recognition, Proceedings of the 27th DAGM Symposium, Vienna, Austria, Berlin, Springer-Verlag (2005) 451–458
12. Monteil, J., Beghdadi, A.: A new interpretation and improvement of the nonlinear anisotropic diffusion for image enhancement. IEEE Transactions on Pattern Analysis and Machine Intelligence **21** (1999) 940–946
13. Weickert, J.: A review of nonlinear diffusion filtering. In ter Haar Romeny, B., Florack, L., Koenderink, J., Viergever, M., eds.: Scale-Space Theory in Computer Vision, Berlin, Springer-Verlag (1997) 3–28
14. Bronstein, I.N., Semendjajew, K.A., Musiol, G., Mühling, H.: Taschenbuch der Mathematik. 5. edn. Verlag Harri Deutsch, Thun und Frankfurt/Main (2001)
15. Barnard, K., Martin, L., Funt, B., Coath, A.: A data set for color research. Color Research and Application **27** (2002) 147–151