

Color line detection

Vinciane Lacroix

**Signal and Image Centre Department, Royal Military Academy, 1000 Brussels, Belgium
E-mail: Vinciane.Lacroix@elec.rma.ac.be*

Abstract

Color line extraction is an important part of the segmentation process. The proposed method is the generalization of the Gradient Line Detector (GLD) to color images. The method relies on the computation of a color gradient field. Existing color gradient are not “oriented”: the gradient vector direction is defined up to π , and not up to 2π as it is for a grey-level image. An oriented color gradient which makes use of an ordering of colors is proposed. Although this ordering is arbitrary, the color gradient orientation changes from one to the other side of a line; this change is captured by the GLD. The oriented color gradient is derived from a generalization from scalar to vector: the components of the gradient are defined as a “signed” distance between weighted average colors, the sign being related to their respective order. An efficient averaging method inspired by the Gaussian gradient brings a scale parameter to the line detector. For the distance, the simplest choice is the Euclidean distance, but the best choice depends on the application. As for any feature extraction process, a post-processing is necessary: local maxima should be extracted and linked into curvilinear segments. Some preliminary results using the Euclidean distance are shown on a few images.

Keywords: color line, color edge, color ordering, Gaussian gradient

1 Introduction

Feature extraction is important for Computer Vision. The argument to use color information for line detection is the same as for color edge detection: some linear features have a much better contrast in the colored than in the luminance image.

Color edge detection has largely been addressed and effective methods are available (see [1], [2] for an overview). Other features such as corners, interest points and lines have mainly been addressed on luminance images, although some of them include color or multi-band information [3], [4], [5]. Lines deserve specific attention because they are important cues that edge detectors often fail to detect. The methods proposed to detect color edges could be simply transposed to line detection, thus edge and lines (dis)similarities are worth summarizing. In grey-level,

*This study is funded by the Belgian Ministry of Defense



Figure 1: The edge orientations in (a) and (b) are opposite while their directions shown in (c) and (d) are the same

an edge *orientation* and *direction* are defined by an angle in the range $]-\pi, \pi]$ and $]-\pi/2, \pi/2]$ respectively (see Figure 1), the orientation pointing towards *higher intensity* values. Although edges have an orientation, this information is hardly used to build contours. On the other hand, in grey-level images the orientation leads to two types of line: “dark” or “bright”. Line detection in grey-level image thus usually involves ridge (bright) and/or valley (dark) detection. Note that “plateau-like” lines are ignored. Both edge and line detection involve a “resolution” or “scale” parameter. The resolution is linked to the minimum distance between the extracted features, but also to the level of noise in the input or smoothness desired for the output. The performance issues concerning edge and line detection are similar [6]: position precision, robustness to noise and computational complexity.

The proposed method for color line detection is made of three modules: the first one computes an “oriented” color gradient field including a smoothing parameter to cope with noise and an efficient implementation to address the computational complexity, the second produces a line strength and direction, and the last one — out of the scope of this article — extracts local maxima and links them into lines.

In Section 2 several strategies for color line detection are analyzed and our choice motivated. Section 3 presents an “oriented” color gradient field. Section 4 explains how the oriented color gradient is used to derive a line strength and direction. Results are shown and discussed in Section 5. Further discussion is provided in Section 6. Conclusions are summarized in Section 7.

2 Choice for a color line detection strategy

Our choice of strategy for color line detection is motivated by exploring several tracks and analyzing their drawbacks.

Color Lines from color edges: if edges may be detected, detecting lines involves only detecting edges at both sides of the line. In practice, two problems occur. First, the edge detection on very thin lines (one pixel wide) may often miss an edge at one side of the line, the edge model assuming some uniformity at each side of the edge. Second, as the aim of line detection is to locate the axis of the line, a necessary non-trivial process is required to match edges at each side of the line.

Color Line Template: masks similar to the generalized Sobel and Prewitt operator

on a 3×3 window or on a 5×5 in color, could be designed and combined as they are for the GVDG [2]. Such an option would convey the same drawbacks as the ones they bring in color edge detection: their sensitivity to noise.

Transform color information to grey-level: a simple approach is to transform the multi-band information into a grey-level image and perform line detection on this image (on the first component of a PCA for example). In Remote Sensing where road detection on high resolution images may rely on line detection, the “spectral-angle difference” between the pixel radiance with respect to some reference has been used at this aim [5]. A black line detection is then performed so that the linear structures having the most similar signature as the reference are extracted. The choice of the reference is a critical issue: an almost optimal choice will produce a double response instead of a unique response if the color at each side of the line are closer to the reference than the color of the line itself. Moreover, as many as reference signatures are needed to detect all “types” of curvilinear structures.

Combining line detection in each plane: detecting lines in each band, then merging the information seems the most straightforward method; see Figure 2 for an example of edge and line norm fusion. Such an approach would however miss a yellow line at the interface of a red and green surface (see Figure 3 (left)) unless it is sought on the Value or Luminance plane, implying a transformation in another space. Which transformations, which planes to consider and how to combine the outputs? A generic but time consuming solution is proposed in [7].

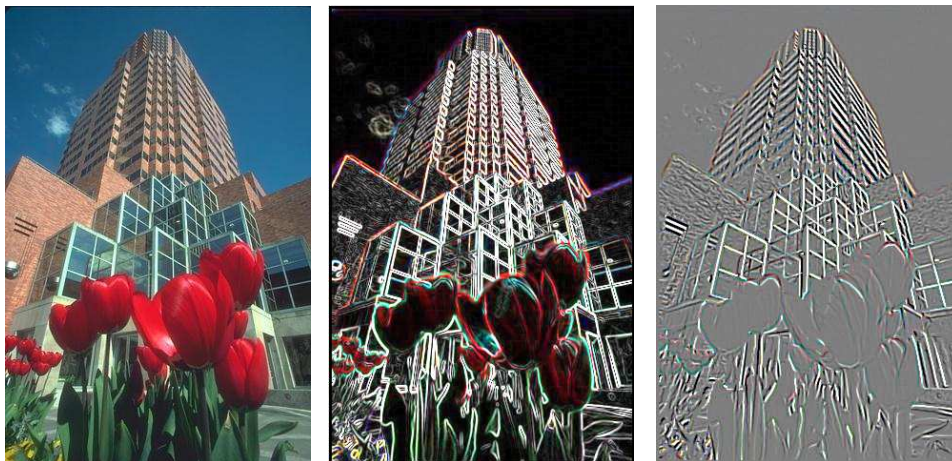


Figure 2: (a) Original Image (b) Edge Fusion: $(R,G,B) = \text{scaled Edge Norm in } (R,G,B)$ (c) Line Fusion: scaled Line Norm in (R,G,B) with grey: no line, brighter: bright line, darker: dark line (GLD computation) [8]

Line detection exploiting local color variation: pixels located on the center of a line are characterized by low color variation along the line and a high variation in the perpendicular direction. If the line lies on a uniform background the color variation seen as a vector in the 3D color space at each side of the line will have

opposite orientation; in the case of a colored line at the interface of two other colors, the orientation will not be opposite, but will vary significantly. Exploring the color variation in a 8-neighborhood will thus give an indication of the presence of a line. Color variation is captured by the color gradient field, but existing color gradient [1], [2], [9], [10] are not “oriented” as they cannot distinguish the passage from a color c_1 to a color c_2 , from its symmetrical (see Figure 3 (right)); we thus introduce an “oriented” color gradient in the next section.

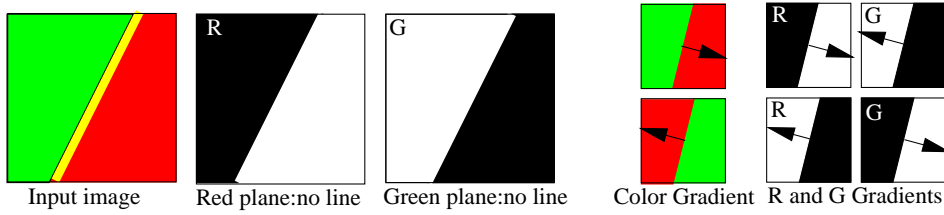


Figure 3: Left: input image on which marginal line output fusion fails; right: an oriented color gradient on two input images and their gradients on Red and Green planes

3 Oriented Color Gradient Field

A color image is seen as a mapping from $Z^2 \rightarrow Z^3$ where each point $p = (i, j)$ of the plane is mapped on a three dimensional vector $\mathbf{c}_p = (r_p, g_p, b_p)$ where $r_p, g_p,$ and b_p represent the red, green and blue values at the coordinates (i, j) . The vector may also be seen as a point (in a three-dimensional space), namely, a “color” point.

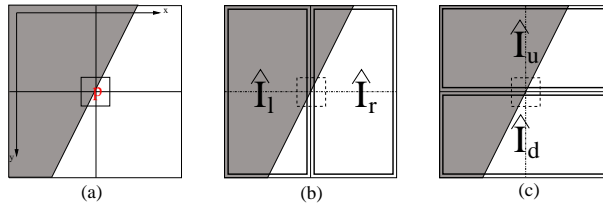


Figure 4: (a) local window at p (b) x component of the gradient: $\hat{I}_r - \hat{I}_l$ (c) y component of the gradient: $\hat{I}_d - \hat{I}_u$

In grey-level images, the intensity variation is captured by $\mathbf{G} = (G_x, G_y)$, the Gradient of the Intensity: at each point, the gradient direction, its norm and its orientation provide the direction of the greatest intensity variation, the amount of variation and the direction of higher intensities respectively; it is the perfect candidate to describe an edge as displayed in Figure 1 (a–b). Each component of the gradient may be seen as the difference between two average intensities [11] or as the **Euclidean distance** between two averages intensities multiplied by a sign representing their respective mathematical order, as seen in Figure 4 where

$\hat{I}_r, \hat{I}_l, \hat{I}_u, \hat{I}_d$ represent the average of the intensity in the zones at the right, left, up and down of the pixel respectively. Thus,

$$G_x(i, j) = s d(\hat{I}_r, \hat{I}_l) \equiv s d_{rl} \quad \text{and} \quad G_y(i, j) = s d(\hat{I}_d, \hat{I}_u) \equiv s d_{du} \quad (1)$$

where $d(a, b)$ is the Euclidean distance and $s = 1$ if $a > b$, $s = 0$ if $a = b$ and $s = -1$ if $a < b$. The norm of the gradient may then be rewritten as:

$$N(i, j) = \sqrt{d(\hat{I}_r, \hat{I}_l)^2 + d(\hat{I}_d, \hat{I}_u)^2} \equiv \sqrt{d_{rl}^2 + d_{du}^2}$$

An oriented color gradient field may thus be obtained by a generalization of Equation 3 from a scalar function I (grey-level) to a vector function \mathbf{A} (color):

$$G_x(i, j) = s d(\hat{\mathbf{A}}_r, \hat{\mathbf{A}}_l) \equiv s d_{rl} \quad \text{and} \quad G_y(i, j) = s d(\hat{\mathbf{A}}_d, \hat{\mathbf{A}}_u) \equiv s d_{du} \quad (2)$$

where $d(a, b)$ is the considered distance and $s = 1$ if $a > b$, $s = 0$ if $a = b$ and $s = -1$ if $a < b$, according to some order relationship. This order is arbitrary (unique if $n = 1$) but all orders will generate similar gradient direction up to π , as the same rule applies to both directions. The lexicographical order is the most natural one. Given a vector $\mathbf{a} = (a_1, \dots, a_n)$ and a vector $\mathbf{b} = (b_1, \dots, b_n)$,

$$\begin{aligned} \mathbf{a} = \mathbf{b} & \quad \text{iff} \quad a_i = b_i \quad \text{for} \quad i = 1, \dots, n. \\ \mathbf{a} < \mathbf{b} & \quad \text{iff} \quad a_1 < b_1 \quad \text{or} \quad a_i = b_i \quad \text{for} \quad i = 1, \dots, j-1 < n \quad \text{and} \quad a_j < b_j \\ \mathbf{a} > \mathbf{b} & \quad \text{iff} \quad a_1 > b_1 \quad \text{or} \quad a_i = b_i \quad \text{for} \quad i = 1, \dots, j-1 < n \quad \text{and} \quad a_j > b_j \end{aligned}$$

Other ordering can be found in literature: color edge detectors based on vector order statistics require an ordering scheme [1]; morphological color edge detectors [10] are a special case of such detectors. However, to our knowledge, none of them uses an *absolute* ordering: in [10], the concept of “extremum” exists but there is no minimum nor maximum; in more general vector order statistics schemes, the concept of “rank” holds, but without privileged orientation.

If the intensity $I(i, j)$ is given by the norm of the vector $\|\mathbf{A}\|$, such as in the RGB representation of an image (i.e. $I(i, j) = \sqrt{A_R^2 + A_G^2 + A_B^2}$), a multi-spectral or hyper-spectral representation, performing an ordering on the norm basis first, and then on each coordinates would be more appropriate.

In the generalization of the norm of the gradient, the orientation disappears:

$$N(i, j) = \sqrt{d(\hat{\mathbf{A}}_r, \hat{\mathbf{A}}_l)^2 + d(\hat{\mathbf{A}}_d, \hat{\mathbf{A}}_u)^2} \equiv \sqrt{d_{rl}^2 + d_{du}^2} \quad (3)$$

If d is the Euclidean distance, it is easily seen that N in Equation 3 is the square root of the marginal squared gradients (i.e. gradient in red, green and blue planes) as already proposed in literature (see [2]). If color perception is an issue, the average intensities $\hat{I}_r, \hat{I}_l, \hat{I}_u, \hat{I}_d$ may be converted in the CIE-L*a*b* space (referred as CIE-Lab in the following), and, when the Euclidean distance is small, other distances such as CIE1994, CIE2000, or CMC could be used [12]. Although there

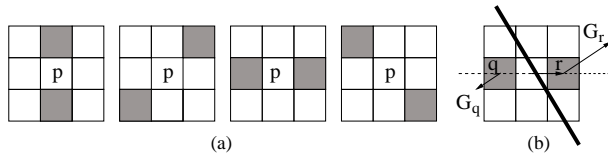


Figure 5: (a) pixels pairs at p (b) gradient vectors configuration for a dark line at p

is still no consensus on the best perceptual distance to use [13], it is recognized that the Euclidean distance in CIE-Lab is *not correct* for small distances (in [3] *the contrary* is assumed). Indeed, similar colors do not hold in spheres in the CIE-Lab space, but rather in ellipsoids, which shapes depend on the color position in the space and on the observer [14].

A two-dimensional color gradient field is thus available on each pixel p , its norm being defined by Equation 2, its direction (including an orientation) defined by the gradient vector in Equation 3. For an efficient computation of the averages, we recommend the weighting factors involved in the Gaussian gradient computation: on each color plane, for the x component of the gradient, a smoothing in the y direction using a Gaussian is performed first (details on the size of the window and the best way to compute the coefficients are given in [8]), then, instead of convolving the resulting image with the derivative of the same Gaussian along x direction as it done for the Gaussian gradient, the derivative mask is divided in a left and right mask, in order to compute a left and right average with all positive coefficients, as the “difference” between the averages is taken out of the computation. The average vector at left and at right may then be obtained by collecting the averages on each color plane. The y component is computed similarly. The scale parameter of the line detector is thus related to the σ of the Gaussian used in this averaging process.

4 Gradient Color Line detection

The grey-level Gradient Line Detector [8] (GLD) exploits the gradient orientation change at each side of the line axis. At each pixel, pairs of opposite pixels in the 8-neighborhood (see Figure 5) producing a negative dot product — the guarantee of an orientation change— are considered, and the square root of the maximum absolute value is taken as line strength. The projection of each gradient along the line joining the pairs enables to distinguish dark lines (lines darker than the background) from bright ones; the method as such cannot detect “plateau” lines.

The oriented color gradient is thus used as input of the GLD, from which “bright” lines or “dark” lines are extracted. In the color context, “bright” and “dark” have to be interpreted with respect to the color order introduced: a line will be detected as bright if its color $c_l > c_b$, where c_l and c_b are the color of line and the background respectively. More precisely, at each pixel p , the oriented color gradient is computed according to Equation 2. In the 8-neighborhood of the pixel

the 4 pairs of symmetrical pixels are considered (see Figure 5). In order to have a line at p , the projection of the oriented color gradient along the line joining some of such pairs should be of different sign. Thus let $\mathbf{d} = (d_x, d_y)$ be the vector joining p to q , where (q, r) is the considered pair. Compute $P_q = G_x(q)d_x + G_y(q)d_y$ and $P_r = -G_x(r)d_x - G_y(r)d_y$. Then if P_q and P_r have the same sign, compute D , the dot product of $\mathbf{G}(\mathbf{q})$ and $\mathbf{G}(\mathbf{r})$:

$$D = G_x(q)G_x(r) + G_y(q)G_y(r)$$

D should be negative, as the orientation of the gradient should change at each side of the line. If P_q and P_r are positive (negative), pixel p is a candidate for a dark (bright) line. Thus if $P_q < 0$ and $P_r < 0$, $D_{bright} = |D|$ else $D_{bright} = 0$; if $P_q > 0$ and $P_r > 0$, $D_{dark} = |D|$, else $D_{dark} = 0$. A pixel may be both candidate for a bright line in some direction (i.e. for some pair (q, r)) and dark in another one. Compute D_{bright} and D_{dark} for each of the 4 pairs and compute respectively L_b and L_d , the maximum of both values. The “bright” and “dark” line strength at p is then $\sqrt{L_b}$ and $\sqrt{L_d}$ respectively. For display purpose, it is convenient to define the “line strength” output at p as $L(p) = \sqrt{L_b}$ if $L_b > L_d$ or $L(p) = -\sqrt{L_d}$ if $L_d \geq L_b$. The direction of the line is given by the direction of the difference of the gradient vectors of the pairs providing the corresponding line strength.

5 Results

The color line filter is characterized by 3 parameters: the smoothing factor σ , the type of distance, and the ordering scheme. It has been applied to two images shown in Figure 6 with slightly different parameters. For the upper image $\sigma = 1$ while for the lower image a value of $\sigma = 0.5$ was necessary to separate some lines. This value is convenient for the upper image: note how the noise in the orange line is smooth out. On the lower image, despite the low value of σ , some lines cannot be separated so that a more precise line detector should be used at some places. All other parameters are similar: the Euclidean distance (d_E) and the lexicographical order (O_{rgb}) were used. The curvilinear structures are well detected in both images.

6 Discussion

A potential undesirable effect of the introduction of an order in the color space is the instability of the edge orientation at the interface of some colors that have a similar color component although their distance is large. Consider the example shown in Figure 7, where one side has the uniform color a , and the other side includes some noise: one pixel has a color b , another one a color b' , with b very close to b' and a far away from b . If the red components of the noisy pixels on the red axis are respectively just lower and greater than the component of a and if the ordering is made on the red component first, then the edge orientation will be

inverted as shown in (b). Note that if another ordering is used (for example starting by the green component first) as in (c), the instability vanishes.

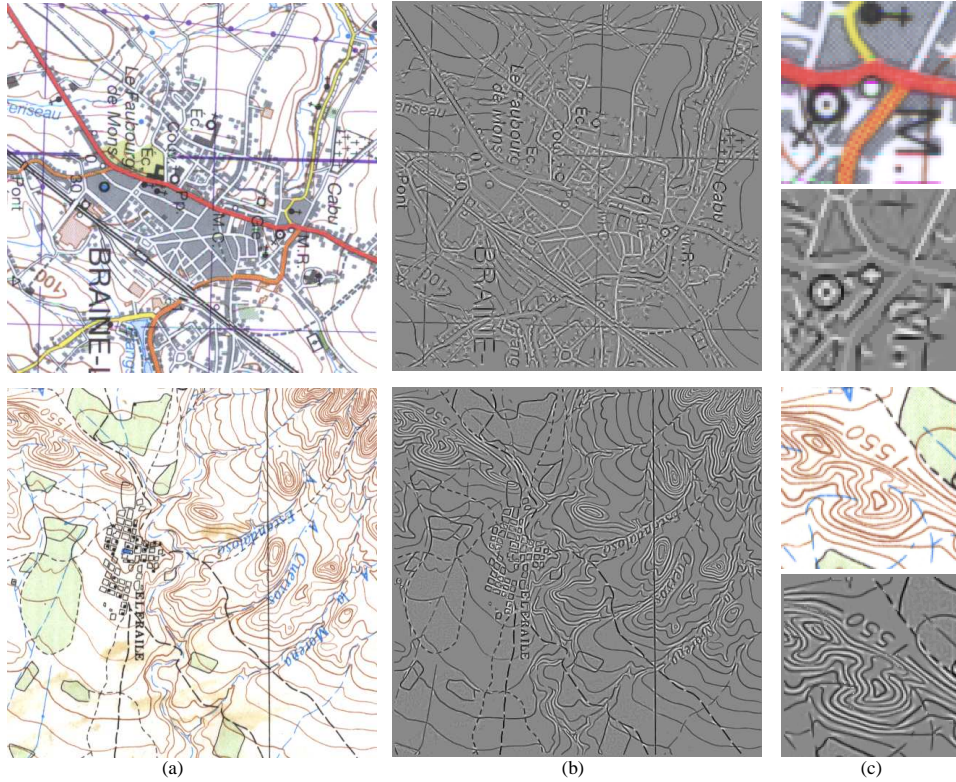


Figure 6: (a) 512×512 color images; (b) Color line filter outputs : grey:0, whiter: “bright” line output, darker: “dark” line output. Filter parameters up: $(1, d_E, O_{rgb})$ down: $(0.5, d_E, O_{rgb})$ (c) zooms on part of the image.

A solution might thus be to detect these specific cases i.e. when \hat{A}_r and \hat{A}_1 or \hat{A}_u and \hat{A}_d are distant while sharing a similar color component, and use a different ordering scheme for the latter. Not more than three ordering scheme are necessary. Of course, the meaning of “dark” and “bright” will change accordingly.

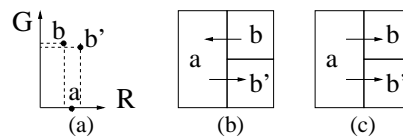


Figure 7: Gradient orientation instability: (a) 3 colors in Color Space having 0 blue components (b) Oriented Gradient using O_{rgb} (c) Oriented Gradient using O_{grb}

7 Conclusions

The introduction of an arbitrary order in the color space enables to produce an oriented color gradient which in turn may be used for color line detection. The proposed color gradient field includes a scale parameter and an efficient implementation based on the Gaussian gradient is proposed. The color gradient computation enables to use other color distances at the price of an additional cost in computer time. The line detection computation is straightforward as it involves computing a few dot products at each pixel. “Bright” and “dark” line strength output are produced, they should be interpreted with respect to the color ordering introduced. The results on some images are encouraging.

References

- [1] Zhu, S-Y. et al., A Comprehensive Analysis of Edge Detection in Color Image Processing, *Optical Engineering*, 38(4): 612-625 (1999)
- [2] Smolka, B. et al. : Noise Reduction and Edge Detection in Color Images, Chapt 4 in *Color Image Processing Methods and Applications*, CRC Press, pp. 75102 (2007)
- [3] Ruzon and Tomasi, Edge, Junction, and Corner Detection Using Color Distributions, *TPAMI* 23(11): 1281-1295 (2001)
- [4] Gevers, T., Van de Weijer, J., Stokman, H.: Color feature detection, Chapt 9 in *Color Image Processing Methods and Applications*, CRC Press, pp. 75102 (2007)
- [5] Christophe, E., Inglada, J.: Robust Road Extraction for High Resolution Satellite Images. *ICIP* (5) pp. 437-440 (2007)
- [6] Canny, J.: A computational approach to edge detection, *TPAMI* Vol.8 (6) pp. 679-697 (1986)
- [7] Stokman, H. at al.: Selection and Fusion of Colour Models for Image Feature Detection, *TPAMI*, 29(3), March, pp. 371-381 (2007)
- [8] Lacroix, V., Acheroy, M.: Feature-Extraction Using the Constrained Gradient. *ISPRS J. of Photogram. and RS*, vol 53, nb. 2, pp. 85-94 (1998)
- [9] Di Zenzo, S., A Note on the Gradient of a Multi-Image, *CVGIP* Vol. 33 (1), pp. 116-125 (1986)
- [10] Evans and Liu, A Morphological Gradient Approach to Color Edge Detection, *IEEE Transactions on Image Processing*, 15(6): 1454-1463 (2006)
- [11] Lacroix, V., A Three-Module Strategy for Edge Detection, *TPAMI* Vol 10 (6), pp. 803 - 810 (1988)
- [12] Ohta, N. and Robertson, A., R., *Colorimetry Fundamentals and Applications*. John Wiley & Sons, Ltd, (2005)
- [13] Kuehni, R., Color difference formulas: An unsatisfactory state of affairs *Color Research & Application* Volume 33, Issue 4, pages 324-326 (2008)
- [14] Berns R, Billmeyer and Saltzman's *Principles of Color Technology*, John Wiley & Sons (2000)