Proc. 34th Annual Workshop of the Austrian Association for Pattern Recognition (AAPR), Zwettl, Austria, pp. 163-170 (May 2010)

Color Space Considerations for Linear Image Filtering

Wilhelm Burger

Upper Austria University of Applied Sciences, Hagenberg, Austria – *wilbur@ieee.org*

Abstract

Filters designed for monochromatic, scalar-valued images are frequently applied to color imagery without considering the vector-valued nature of the data and the effects of nonlinear component values. This paper analyzes the implications of component-wise linear filtering upon the underlying color space and the related chromatic errors that may be unacceptable for high-quality color imaging. It suggests that perceptual uniformity and a careful choice of the working color space are important even for common smoothing filters and similar operations.

1. Introduction

Most images today are color images and filtering them is an everyday routine. Typically, linear and non-linear filters designed for grayscale images are used for color images by simply applying them to the individual color channels, i. e., by executing a scalar technique on vector-valued data. Although the resulting color artifacts appear to be acceptable for the majority of viewers, the errors introduced by these methods may be substantial and even unacceptable for high-quality color imaging. In this context, the photometric properties and, in particular, the potential non-linearities of the underlying color space are often ignored. For example, most color images produced today by digital cameras and scanners are in sRGB ("Standard RGB") or a similar format, characterized by gamma-mapped (i. e., strongly non-linear) color components for the purpose of high coding efficiency and simplicity of use [9].

The results of linear filtering depend on the color space being used, but most image editing tools (including professional software) nevertheless apply filters directly to non-linear (sRGB) color components, without previous linearization or color space conversion. Not only does a linear operation in a non-linearly distorted space seem incorrect by intuition but it actually introduces substantial errors and visually disturbing results, as described below. Which is the correct color space to use? Should the operation be performed in linear RGB, CIEXYZ, or in a perceptually uniform color space such as CIELAB? Although the issues related to color mixtures and interpolation have been investigated for a long time (see, e. g., [4, 14]), their relevance in the context of image filtering has received only limited attention in the literature, with notable exceptions [1, 5, 10, 11].

In the following, we investigate the nature of scalar, linear filters for color images and derive requirements for the results, based on colorimetric and perceptual arguments. Four popular color spaces are evaluated: non-linear sRGB [9], linear RGB, CIELUV, and CIELAB [8]. We do not consider cylindrical color spaces (such as HSV or HLS) here because they lack a precise photometric specification. We first describe the implications of applying scalar linear filters to vector-based data and the implicit assumptions about the underlying color space. Next, we demonstrate how these assumptions are violated by processing images represented in non-linear and "light-linear" RGB coordinates. We subsequently illustrate that "correct" (visually acceptable) results are not obtained by relying on the *physical* but the *perceptual* properties of the working color space. Examples are shown to demonstrate these effects.

2. Scalar linear filters applied to vector-valued images

Given a discrete *scalar* (grayscale) image $I(u,v) \in \mathbb{R}$, the application of a linear filter can be expressed as a linear 2D convolution

$$\bar{I}(u,v) = [I * H](u,v) = \sum_{(i,j)\in R_H} I(u-i, v-j) \cdot H(i,j), \tag{1}$$

where H denotes a discrete filter kernel defined over the (usually rectangular) region R_H, with $H(i,j) \in \mathbb{R}$. For a *vector*-valued image $\boldsymbol{I} = (I_1, I_2, \ldots, I_n)$ or $\boldsymbol{I}(u,v) \in \mathbb{R}^n$, with n (color) components I_k, the above linear filter can be written as $\bar{\boldsymbol{I}}(u,v) = [\boldsymbol{I} * H](u,v)$, with the filter kernel H still being scalar-valued. The k^{th} element of the result vector, $\bar{I}_k(u,v) = [I_k * H]\,(u,v)$, is simply the ordinary scalar convolution (1) applied to the corresponding component plane I_k. In case of a RGB color image (with $n = 3$ components), the filter kernel H is applied separately to the scalar-valued I_R, I_G, I_B planes, i. e.,

$$\bar{\boldsymbol{I}}(u,v) = \begin{pmatrix} \bar{I}_R(u,v) \\ \bar{I}_G(u,v) \\ \bar{I}_B(u,v) \end{pmatrix} = \begin{pmatrix} [I_R * H]\,(u,v) \\ [I_G * H]\,(u,v) \\ [I_B * H]\,(u,v) \end{pmatrix}, \tag{2}$$

which is how linear filters for color images are typically implemented in practice.

Linear smoothing filters

Let $\mathcal{R}_{u,v} = \{\boldsymbol{c}_1, \ldots, \boldsymbol{c}_K\}$ denote the set of color vectors contained in the support region of the kernel H at a given position (u,v) in the original image $\boldsymbol{I}$, where K is the size of H. With arbitrary kernel coefficients $H(i,j) \in \mathbb{R}$, each resulting color vector $\bar{\boldsymbol{I}}(u,v) = \bar{\boldsymbol{c}}$ is a linear combination of original color vectors $\mathcal{R}_{u,v}$. If the kernel is normalized, i. e., $\sum_{(i,j)} H(i,j) = 1$, the result is an affine combination of the original colors. In case of a typical smoothing filter, with all coefficients $H(i,j)$ being positive and normalized, each resulting color vector $\bar{\boldsymbol{c}}$ is a *convex combination* of the original color vectors $\boldsymbol{c}_1, \ldots, \boldsymbol{c}_K$.

Figure 1 shows the result of filtering a synthetic test image with a normalized Gaussian kernel with radius $\sigma = 3$. Included are the grayscale images obtained from the corresponding *luminance* (Y) values (see Sec. 3.). The colors inside the horizontal bar at the center of the test image are 50% desaturated but have the same luminance as their neighboring colors. The bar should thus not show in the filtered images, unless incorrect luminance values are produced. This happens in the sRGB results (Fig. 1(b)), which also show strong dark bands, particularly along the red-blue, magenta-blue, and magenta-green edges.

Response to a color step edge

Assume, as a special case, that the original RGB image $\boldsymbol{I}$ contains a *step edge* separating two regions of constant colors $\boldsymbol{c}_1 = (R_1, G_1, B_1)$ and $\boldsymbol{c}_2 = (R_2, G_2, B_2)$, respectively, as illustrated in Fig. 2(b). If placed at a position (u,v), where the normalized smoothing kernel H is fully supported by elements of constant color $\boldsymbol{c}_1$ (with the kernel is not overlapping any color edge), the trivial response of the filter is $\bar{\boldsymbol{I}}(u,v) = \sum \boldsymbol{c}_1 \cdot H(i,j) = \boldsymbol{c}_1 \cdot \sum H(i,j) = \boldsymbol{c}_1 \cdot 1 = \boldsymbol{c}_1$, i. e., the unmodified, original color $\boldsymbol{c}_1$. Alternatively, if the filter kernel is placed at some position (u', v') *on* the step edge, some of its coefficients ($\mathcal{R}_1$) are supported by pixels with color $\boldsymbol{c}_1$, while the remaining coefficients ($\mathcal{R}_2$) overlap pixels with color $\boldsymbol{c}_2$. Since $\mathcal{R}_1 \cup \mathcal{R}_2 = \mathcal{R}$ and the kernel is normalized, the resulting color at position (u', v') is

$$\boldsymbol{c}'_\lambda = s_1 \cdot \boldsymbol{c}_1 + s_2 \cdot \boldsymbol{c}_2 = s_1 \cdot \boldsymbol{c}_1 + (1 - s_1) \cdot \boldsymbol{c}_2 = \boldsymbol{c}_1 + \lambda \cdot (\boldsymbol{c}_2 - \boldsymbol{c}_1), \tag{3}$$

Proc. 34th Annual Workshop of the Austrian Association for Pattern Recognition (AAPR), Zwettl, Austria, pp. 163-170 (May 2010)

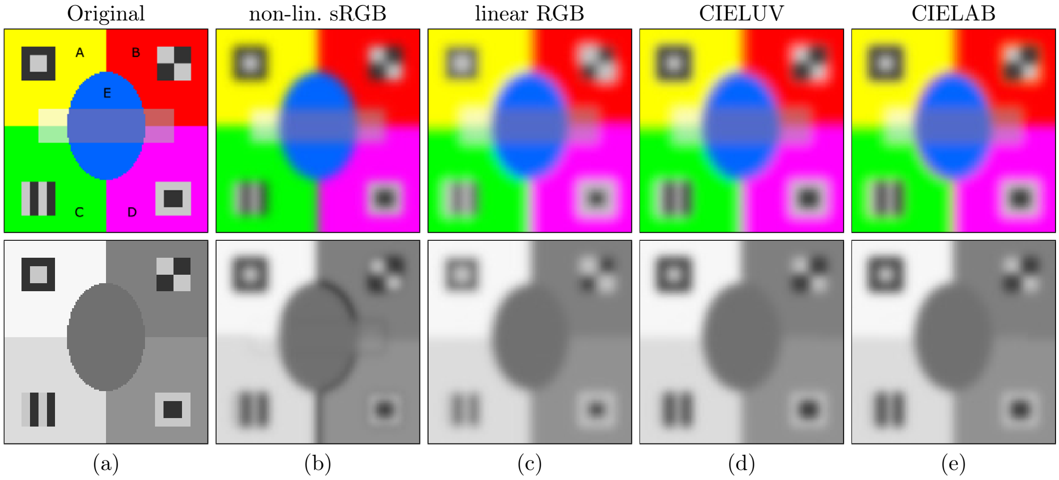

Figure 1. Gaussian smoothing performed in different color spaces. Synthetic test image (a); Gaussian smoothing filter applied to sRGB (*nonlinear*) color components (b), RGB (*linear*) components (c), CIELUV (d) and CIELAB components (e). The bottom row shows the corresponding luminance (Y) images. Images are available at electronically at `http://staff.fh-hagenberg.at/burger/`. Note that the final appearance depends strongly on display or printer characteristics and some of the effects may not show as described.

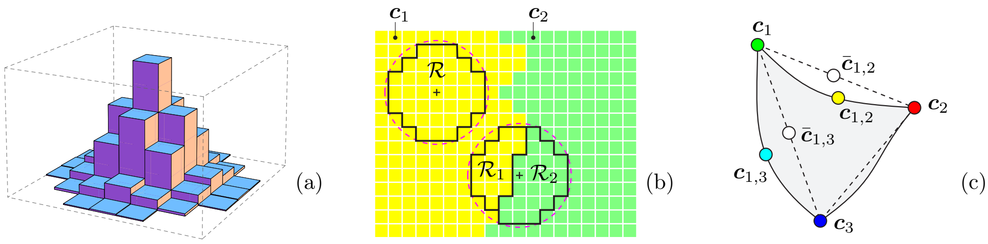

Figure 2. Discrete 2D Gaussian filter kernel with radius 3 and support region $\mathcal{R}$ (a). Filter kernel positioned over a region of constant color $\boldsymbol{c}_1$ and over a color step edge $\boldsymbol{c}_1 \rightarrow \boldsymbol{c}_2$, respectively (b). Different results are obtained by linear interpolation in a "distorted" color space (c).

for some $\lambda \in [0, 1]$, which is a point on the straight line segment between the color coordinates $\boldsymbol{c}_1, \boldsymbol{c}_2$. Thus, at a step edge between two colors, the transient colors produced by a (normalized) smoothing filter at different positions always correspond to points on the straight line between the two original color vectors. As an example, Fig. 3(a) shows the results of linear interpolation between pure red and green, calculated in four different color spaces. Note that, although the individual RGB components run over the same range of values, the actual colors produced differ strongly (also see Fig. 4(d)).

This means that, in general, the intermediate colors produced by a linear smoothing filter are contained within the *convex hull* of the set of contributing colors in the working n-dimensional color space, regardless *which* color space is used. The relationship between linear, scalar convolution and convex linear combinations thus implies that the underlying color space is metric and the triangle inequality $\|\boldsymbol{c}_1 + \boldsymbol{c}_2\| \leq \|\boldsymbol{c}_1\| + \|\boldsymbol{c}_2\|$ holds under the Euclidean norm. The question, if the intermediate chromaticities produced by component-wise filtering are "correct", cannot be answered from a mathematical but only from a perceptual point of view. The CIELUV and CIELAB color models were explicitly designed to provide perceptual uniformity over a large color range and facilitate the use of the Euclidean norm as a valid distance measure between