

# A Multiscale Framework for Spatial Gamut Mapping

Ivar Farup, Carlo Gatta, and Alessandro Rizzi

**Abstract**—Image reproduction devices, such as displays or printers, can reproduce only a limited set of colors, denoted the color *gamut*. The gamut depends on both theoretical and technical limitations. Reproduction device gamuts are significantly different from acquisition device gamuts. These facts raise the problem of reproducing similar color images across different devices. This is well known as the *gamut mapping* problem. Gamut mapping algorithms have been developed mainly using colorimetric pixel-wise principles, without considering the spatial properties of the image. The recently proposed multilevel gamut mapping approach takes spatial properties into account and has been demonstrated to outperform spatially invariant approaches. However, they have some important drawbacks. To analyze these drawbacks, we build a common framework that encompasses at least two important previous multilevel gamut mapping algorithms. Then, when the causes of the drawbacks are understood, we solve the typical problem of possible *hue shifts*. Next, we design appropriate operators and functions to strongly reduce both *haloing* and possible undesired *over compression*. We use challenging synthetic images, as well as real photographs, to practically show that the improvements give the expected results.

**Index Terms**—Gamut, gamut mapping, haloing, hue shift, multiscale, spatially variant.

## I. INTRODUCTION

A COLOR gamut is the set of all colors reproducible by a given device or present in a given image. When reproducing color images, we have to deal with the problem of incompatible color gamuts. Most probably, some colors present in the given image are not within the color gamut of the reproduction medium. Thus, in order to be able to reproduce the original image properly, we have to modify at least some of its colors, i.e., perform gamut mapping (GM). Solving this problem means searching for a tradeoff between accuracy and pleasantness of reproduction, cf., e.g., the rendering objectives according to Hunt [1], or the rendering intents as defined by the International Color Consortium (ICC) [2].

Eschbach [3] pointed out that the reproduction accuracy of an image is not well defined, although the accuracy of the reproduction of a single color is. The very fact that colorimetrically less accurate GM algorithms (GMAs) give visually more “accurate” results [4], shows that other aspects than pure colorimetry play important roles in image reproduction. Recently, Bala [5]

Manuscript received December 20, 2007; revised June 4, 2007. This work was supported by the PRIN-MIUR Research Project 2005115173-002. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Gaurav Sharma.

I. Farup is with the Gjøvik University College, Gjøvik N2802, Norway.

C. Gatta and A. Rizzi are with the University of Milano, Milano, Italy.

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TIP.2007.904946

emphasized the importance of other characteristics (which he called *dimensions*) in color imaging, including the spatial, the spectral, and the goniometric dimensions.

A recent trend in the GMA research is the use of spatial information to compute the pixels’ color transformation from input (image) to output (device) gamut. To achieve this, low-/high-pass filtering or pyramidal decomposition is often used. Unfortunately, these techniques share the problem of generating halos close to sharp boundaries, induced by the spatial computation itself. Another problem of such spatial GMAs is the hue shift induced from compressing a group of pixels towards the gray using the same vector in the chosen color space, thus erroneously mapping colors that don’t need to be changed. This aspect is extensively discussed in the paper.

In this work, we present a framework for multiscale spatial GMA. In developing a robust framework, we focused our attention on two common problems: local hue shifts and haloing artifacts. We show a general method to completely avoid hue shift commonly caused by the multilevel method. For reducing the haloing artifacts, we show a general approach that takes into account the local properties of a pixel in the color space, thus enabling the proposed algorithm to avoid unwanted compressions or expansions. The methods we present can easily be incorporated in other multilevel GMAs. They can also be extended by the use of alternative color spaces or alternative scale-space representations.

In Section II, the state of the art of spatial GM is summarized and, in Section III, one possible formulation of a multiscale GMA based on the established ideas is presented. Then, in Section IV, using the formulation of Section III, the advantages and drawbacks of the simple multiscale GMA is discussed in detail. Motivated by the analysis of the advantages and drawbacks, we propose a new multiscale framework for spatial GM in Section V. Section VI discusses some of the properties of the proposed method, and results for real images are given in Section VII. Finally, possible extensions of the method are elaborated in Section VIII.

## II. BACKGROUND AND STATE OF THE ART

Although GMAs has been an active field of research for a long time, most of the activity has been related to simple mappings in color space, not taking the spatial dimension into account [4]. Much of the traditional research consists in proposing new mappings in the CIELAB, CIECAM02, or a related color space, and then performing panel tests in order to evaluate the efficacy of the proposed method.

The first attempt at a spatial GMA known to the authors, was made by Meyer and Barth [6]. The first step of their algorithm is lightness compression using low-pass filtering in the Fourier spatial frequency domain. The dynamic range of the low-pass-filtered image is then compressed to that of the reproduction

medium and the high-pass-filtered image detail information is added back to it.

Bala *et al.* [7], [8] introduced a two-level approach for spatial GMAs related to the ideas of Meyer and Barth [6]. The image is first gamut clipped, mapping out-of-gamut colors to the nearest surface point of the same hue. Then, the difference between the luminance of the original and the gamut-mapped image is high-pass filtered and added to the gamut-mapped image, since gamut clipping typically reduces the high-frequency content of the image. Finally, the image is gamut clipped, mapping out-of-gamut colors to the surface in a direction towards a neutral point whose luminance is that of the cusp color [4]. Later, the method was improved by applying different algorithms onto small and large objects [9].

A multiscale spatial GMA was recently developed by Mroovic and Wang [10]. The original image is decomposed into different spatial frequency bands, then lightness compression and GM are applied to the lowest frequency image. Subsequently, the next highest frequency band is added to the resulting image, and the process iterated until all frequencies are treated. The effect is that the intra-image differences in the original image are well maintained, although the overall color difference between the original and the gamut-mapped image is greater than that produced by conventional mappings in color space. Different GM strategies can be applied for the different frequency bands, and different low-pass filters can be selected. Thus, this is quite a modular method.

A multiresolution GM strategy has been suggested by McCann [11]. The algorithm is based upon the Retinex theory of color vision [12], [13]. The algorithm works at different scales, starting at the smallest subsampled image, and performs mapping in a way that tries to preserve ratios between the values of neighboring pixels as much as possible.

Very recently, a new spatial gamut mapping algorithm was proposed by Zolliker and Simon [14]. They followed the approach by Bala *et al.* [7], [8], but instead of Gaussian filtering, they applied bilateral filtering [15], thus avoiding unwanted haloing artifacts near sharp boundaries.

A completely different approach was taken by Nakauchi *et al.* [16]. They defined GM as an optimization problem of finding the image that is perceptually closest to the original and has all pixels inside the gamut. The perceptual difference was calculated by applying band-pass filters to Fourier-transformed CIELAB images and then weighing them according to the human contrast sensitivity function. Thus, the best gamut-mapped image is the image having contrast (according to their definition) as close as possible to the original.

Recently, Kimmel *et al.* [17] presented a variational approach to spatial gamut mapping. Their treatment starts with the presentation of a new fitness measure, closely related to a recent measure proposed for Retinex [18]. It is shown that the gamut mapping problem leads to a quadratic programming formulation, guaranteed to have a unique solution if the gamut of the target device is convex.

Apart from the works of Nakauchi *et al.* [16] and Kimmel *et al.* [17], all of the above authors introduce spatial dependency into the GMAs by using a multilevel approach, sometimes with

only two levels. Subsampling as a way to obtain spatial processing was proposed approximately simultaneously by Burt [19] and Crowley [20]. They argued that human beings look at things on many spatial scales at the same time, and that computer algorithms should in some sense mimic this ability. They, thus, introduced the concept of subsampled image pyramids, also commonly referred to as MIP maps [21].

Another way to reach a multilevel approach is to introduce a scale space [22]. In the scale-space formalism, the images are not subsampled, but instead low-pass filtered in order to get access to the spatial properties of the image at different scales. In this paper, we refer to spatial GM based upon the pyramidal decomposition as *multiresolution GM*, whereas spatial GM based upon a scale-space representation will be referred to as *multiscale GMAs*. By *multilevel GM* we mean both.

### III. MULTISCALE GMA

McCann has shown that changing all the colors of an image in the same way is significantly less perceivable than changing different colors in different ways [23]. This happens because human visual system perceives spatial ratios more than absolute values. For a spatial GMA context, this means that we should treat large areas of the image as much as possible in the same way in order to maintain the local relationships between colors. For example, if a large area of the image is in average out of gamut, the whole area should be gamut compressed in order to preserve the local relationship among colors. However, single pixels that are out of gamut could be treated independently, e.g., by gamut clipping.

Such behavior can be achieved by a multiscale approach: First, the image is low-pass filtered using a large convolution kernel. The low-pass-filtered image is then mapped using a spatially invariant GMA. The difference between the gamut-mapped low-pass-filtered image and the low-pass-filtered image is then added back to the original image. Since the resulting image can still have out-of-gamut pixels in higher frequency bands, the process is iterated with decreasingly smaller convolution kernels. The whole process can best be described as an iterative multiscale GMA defined in terms of the spatial invariant GMA and the low-pass filters chosen. Out-of-gamut regions will be detected and mapped at the stage where the whole region in average (in the sense of the chosen low-pass filter) is out of gamut.

#### A. Unified Viewpoint for Existing Multiscale GMAs

In this section, we build a mathematical formulation of the multiscale GMAs briefly introduced above. The aim of this section is to develop a unified viewpoint for the analysis of the advantages and drawbacks of these approaches.

Formally, since we will need to treat image pixels as vectors in the color space, we describe the image as a vector-valued function of the position in the image, i.e.,  $\mathbf{f} : \mathcal{I} \rightarrow \mathbb{R}^3$ , where  $\mathcal{I} = \{0, \dots, W-1\} \times \{0, \dots, H-1\}$  is the spatial image domain, and  $\mathbb{R}$  is the set of all real numbers,  $W$  and  $H$  being, respectively, the width and the height of the image measured in numbers of pixels. For simplicity, we group all possible images into an abstract functional space  $\mathcal{F}$ ,  $\mathbf{f} \in \mathcal{F}$ . We need one operator for the spatial invariant GMA,  $G : \mathcal{F} \rightarrow \mathcal{F}$ , and a sequence

of  $P \in \mathbb{N}^+$  low-pass filtering operators  $(L_p)_{p=0}^{P-1}$ ,  $L_p : \mathcal{F} \rightarrow \mathcal{F}$ ,  $\mathbb{N}^+$  being the set of positive natural numbers.<sup>1</sup> The last low-pass filter is taken to be the identity operator,  $L_{P-1} = I$ , in order to make sure that the final step of the iterative procedure is a spatially invariant GMA. In this way, all colors of the final image are guaranteed to be inside the destination gamut.

Denoting by  $\mathbf{f}_0$  the original input image, the iterative scheme can be summarized in the following equation for  $p \in \{0, \dots, P-1\}$ :

$$\mathbf{f}_{p+1} = \mathbf{f}_p + G(L_p(\mathbf{f}_p)) - L_p(\mathbf{f}_p). \quad (1)$$

The final gamut-mapped image is  $\mathbf{f}_P$ . In order for this to be meaningful, it is important that a sufficiently perceptually homogeneous color space is used.

It is easy to show that formula (1) incorporates the multiscale GMAs proposed by Bala *et al.* [7], [8] (denoted with the apex  $B$ ) by rewriting it as follows, for the case of two levels:

$$\mathbf{f}_1^B = G(L_0(\mathbf{f}_0^B)) + \underbrace{\mathbf{f}_0^B - L_0(\mathbf{f}_0^B)}_{\text{High-pass filter}}. \quad (2)$$

We can also show that (1) is conceptually similar to the Morovic and Wang [10] approach (denoted with the apex  $M$ ). In our understanding, their GMA can be formulated as follows:

$$\mathbf{f}_{p+1}^M = G(\mathbf{f}_p^M) + \underbrace{L_{p+1}(\mathbf{f}_0^M) - L_p(\mathbf{f}_0^M)}_{\text{Band-pass filter}} \quad (3)$$

and (1) can easily be rewritten as

$$\mathbf{f}_{p+1} = G(L_p(\mathbf{f}_p)) + \underbrace{\mathbf{f}_p - L_p(\mathbf{f}_p)}_{\text{High-pass filter}}. \quad (4)$$

The formulations (3) and (4) are conceptually very similar, since a high-pass filter combined with a low-pass filter in the next stage of the iterative approach (4) is effectively a band-pass filter of the input image (3).

The above formulation of multilevel GM turns into a conventional spatially invariant GM in the trivial case where  $P = 1$  and the low-pass filter is the identity operator,  $L_0 = I$ .

### B. Specific Implementation of a Multiscale GMA

To obtain a specific implementation of (1), we have to choose the operators  $G$  and  $(L_p)_{p=0}^{P-2}$ . We choose  $G = G_{\text{cl}}$ , a pure gamut clipping along straight lines towards the gamut center in the CIELAB color space (CIE L\*a\*b\* 1976 with D50 white point [24]). To implement clipping, the gamut boundary is determined using the modified convex hull method (with  $\gamma = 0.2$ ) suggested by Balasubramanian [25], since it was recently demonstrated that this method is particularly well suited for this purpose by Bakke *et al.* [26].

Since we will use a discrete scale variable, the low-pass filtering is performed with Gaussian kernels [27]

$$g_{\sigma_p}(x, y) = \frac{1}{2\pi\sigma_p^2} \exp\left(-\frac{x^2 + y^2}{2\sigma_p^2}\right). \quad (5)$$

<sup>1</sup>Actually, since  $G$  is spatially invariant, it is a pixelwise operation that could be defined in terms of a function  $g : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ , such that  $G(\mathbf{f})(x, y) = g(\mathbf{f}(x, y))$ .

We build a discrete scale space by halving the standard deviation  $\sigma_p$  at each level (also referred to as an octave space) [27]. In order to make sure that  $\sigma_{P-2} = 1$ , we put  $\sigma_p = 2^{P-2-p}$ , hence

$$L_p(\mathbf{f}) = g_{2^{P-2-p}} * \mathbf{f} \quad (6)$$

where  $*$  denotes convolution. The number  $P$  of low-pass filters is chosen according to the size of the image. The size of the largest scale convolution kernel is approximately  $4\sigma_0$ . Since the largest convolution kernel should be significantly smaller than the whole image but still larger than the important details in the image, we choose to have

$$P = \lceil \log_2(\max(W, H)) \rceil - 2. \quad (7)$$

where  $\lceil \cdot \rceil$  is the ceiling function. This choice of  $P$  is quite arbitrary; and it is not important for the performance of the resulting algorithm. However, too large  $P$  values can lead to artifacts near the edges of the resulting image. Since the size of the largest convolution kernel is of the same order as the image size, the filtering is performed in the Fourier domain for computational efficiency.

The choice of Gaussian kernels for constructing the scale space is not mandatory for the proposed framework. Alternatives are discussed in Section VIII.

## IV. MOTIVATION: ADVANTAGES AND DRAWBACKS OF MULTILEVEL GMA

It is widely accepted that a multilevel GMA like the one described above can perform better than or equal to spatially invariant GMAs [8], [10]. This is because multilevel GMAs can take into account the spatial properties of the input image, and not only the input gamut. Moreover, a multilevel approach can treat the same color differently, according to frequency content.

In this section, we highlighted advantages and drawbacks of multilevel GMAs. The main motivation of this paper is to identify drawbacks and propose general approaches and methods to avoid or reduce these drawbacks. In the following sections, we will discuss this using the multilevel GMA defined in the previous section.

### A. Advantages

The main advantage of the multilevel approach over the spatially invariant ones is a better *rendering of details*. In spatial invariant GM, one has to deal with the trade off between gamut clipping and gamut compression. If the image is gamut clipped, it retains much of the contrast and vividness of the original, at the cost of losing details. Gamut compression, on the contrary, attenuates contrast and vividness while retaining details. Intermediate solutions are of course possible, e.g., using knee or sigmoidal compression functions [4]. With multilevel GMAs, it is possible to obtain compression of image regions with many details while clipping regions of constant color, thus avoiding unnecessary desaturation [8], [10].

### B. Drawbacks

The multilevel algorithm smoothly compresses differently sized areas of the image by compressing progressively from

lower to higher spatial frequencies. The compression at a fixed level (i.e., at a specific spatial frequency) is performed using a spatially invariant GMA. The spatial computation is solely due to the frequency filtering, and it's not performed again when gamut mapping a single pixel color. The compression at one level is then propagated to upper levels towards higher spatial frequencies [cf. (1)].

The mapping of a pixel color at a fixed level (except for the last step) is, thus, the result of a weighted spatial average of the image at the level below [cf. (6)]. Consequently, when the gamut compression is propagated from the lower to the higher levels, each individual pixel is gamut mapped as if its pixel value equaled the (weighted) mean pixel value of its neighborhood. At first sight, this seems reasonable and perfectly in accordance with the multiscale theory of image processing. However, treating every color as the mean of its neighborhood unfortunately can generate different types of artifacts depending both on the spatial and color distribution in the image. In particular, outlier pixels involved in the averaging process can be subjected to *hue shift*, *hue change*, and *gamut expansion*. Moreover, these phenomena are strongly interrelated, and they can also cause *halos* in the resulting gamut-mapped image.

**1) Local Hue Shift, Hue Change and Gamut Expansion:** To illustrate how these three drawbacks are originated by the above described method, we build a synthetic image ( $8 \times 8$  pixels), and then show what can happen in a 2-D simplified case. First, we color every pixel with  $(L^*, a^*, b^*) = (66.5, 70, 20)$ , then we add Gaussian noise with a standard deviation of  $\sigma = 5$  to the  $(a^*, b^*)$  values. Then, we change three single pixels of the image using the following values:  $(L^*, a^*, b^*) = (66.5, 5, 45)$  (pixel *a*),  $(L^*, a^*, b^*) = (66.5, 10, 3)$  (pixel *b*), and  $(L^*, a^*, b^*) = (66.5, -25, -15)$  (pixel *c*). Since we only change three out of the 64 pixels, the mean value of the image is still very close to  $(66.5, 70, 20)$ . Thus, in the multilevel framework, all pixels in the image will be mapped as if their pixel value were  $(66.5, 70, 20)$ .

Fig. 1 shows the cross section of the CIELAB color space with  $L^* = 66.5$ . The black curve is the boundary of the gamut of the ISO uncoated ICC profile [28] (this is the destination gamut for all gamut-mapped images presented in the paper). The small dots are pixel colors, the big dot is the average color of the image, the vector indicates the clipping of the average color towards the gamut center.

Fig. 2 shows what happens when we, in accordance with the multilevel method described above, apply the clipping vector of the mean to all colors present in the image [see (1)]. This is what typically happens at each level in the multilevel scheme. Most of the pixels of the cloud do not move on constant hue lines (that would be straight lines towards the origin). Even though the induced hue shifts are small, it is not what is intended. If the cloud represents noise, the noise will be more pronounced in the gamut-mapped image since relative hue differences are increased. A significantly larger hue shift effect can be noticed for the color *a*: In this unfortunate case, the hue angle shift is about  $32^\circ$ . Another case is represented by the color *b*: the hue is completely changed as a consequence of crossing of the gray axis. The case of color *c* is symptomatic of an even bigger problem: In-gamut colors can be mapped outside the gamut causing an unwanted gamut expansion. This fact could erroneously be con-

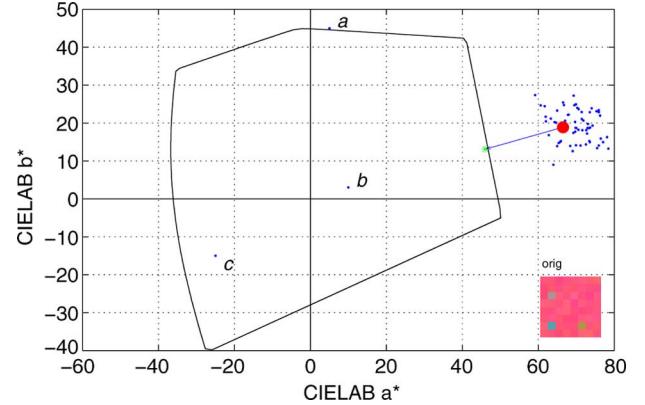


Fig. 1. First synthetic image viewed in the  $(a^*, b^*)$  plane of the CIELAB color space for  $L^* = 66.5$ . The small dots are the individual pixels, mainly forming a cloud. The three pixels marked *a*, *b*, and *c* are not a part of the cloud, as discussed in the text. The big filled circle is the average color of all the pixels, and the small filled circle on the boundary is the closest in-gamut color having the same CIELAB hue ( $h_{ab}$ ) as the average color. The synthetic image is shown in the lower right corner.

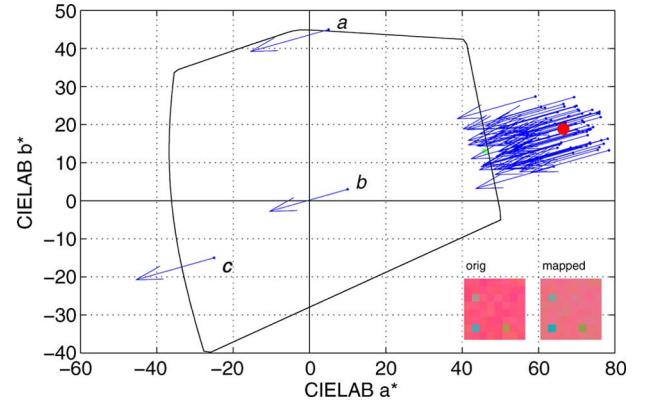


Fig. 2. Result of applying the same color change to all pixels of the synthetic image of Fig. 1. The synthetic image and the resulting gamut-mapped image are shown in the lower right corner.

sidered a minor problem since a multilevel algorithm will take care of this out-of-gamut pixel in a higher level. However, we show in the next section that this can cause a strong haloing effect.

**2) Haloing:** The synthetic image of the previous section provides a clear example that the above described method changes colors in a way that does not preserve local relationships (see, e.g., pixel *b* with respect to cloud pixels). This implies that colors are modified in a way that is not consistent with color appearance prescriptions [11]. To understand how this erroneous mapping can generate haloing problems, we constructed a second synthetic image ( $512 \times 512$  pixels) with challenging properties. The left part of Fig. 3 shows the synthetic image: it is composed of several patches whose  $(L^*, C_{ab}^*, h_{ab})$  values are given in Table I.

Between the patches, there are very sharp edges that can cause haloing problems. The four small squares share the hue angle with the saturated ones in which they are contained. The right part of Fig. 3 shows the result of the standard multilevel GMA [formula (1), seven scales] on the synthetic image. Fig. 4 shows  $L^*$  and  $h_{ab}$  values of a scan-line (the horizontal white line of

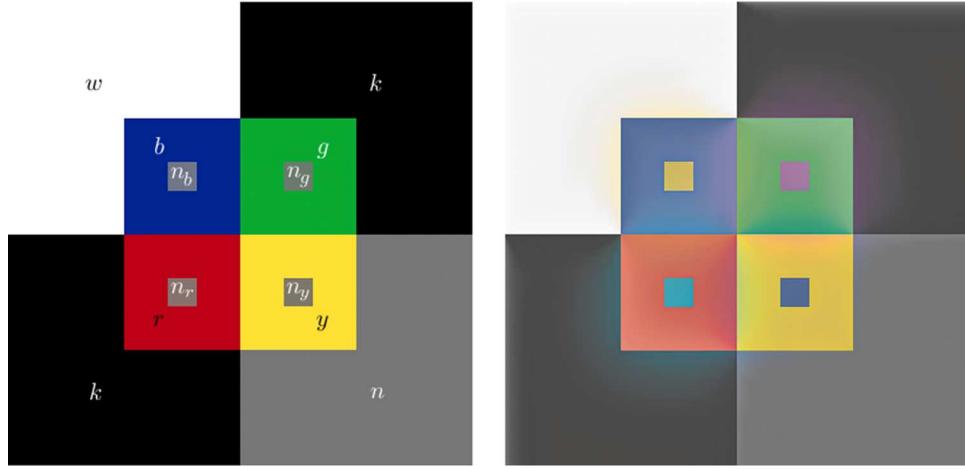


Fig. 3. (Left) Second synthetic image with labels indicating the individual colors as listed in Table I, and (right) the result of gamut mapping using the multiscale method (1) mapping to the ISO uncoated gamut.

TABLE I  
LCh VALUES FOR THE SYNTHETIC IMAGE (LEFT PART OF FIG. 3)

Patch	$L^*$	$C_{ab}^*$	$h_{ab}$
w	100	0	0
n	50	0	0
k	0	0	0
r	40	100	$\pi/5$
g	60	100	$4\pi/5$
b	10	100	$3\pi/2$
y	90	100	$\pi/2$
$n_r$	50	10	$\pi/5$
$n_g$	50	10	$4\pi/5$
$n_b$	50	10	$3\pi/2$
$n_y$	50	10	$\pi/2$

Fig. 3) of both the input and gamut-mapped synthetic image. The dashed line is the input image and the solid line is the gamut-mapped one. We plot  $L^*$  and  $h_{ab}$  values to show the strength of halos both in  $L^*$  and  $h_{ab}$  channels to highlight that the halo is colored. While halos are not heavy in the  $L^*$  channel, it is evident that there is a big hue shift (from pixel 0 to 128) because this part of the input image is achromatic (for convenience, we set  $h_{ab} = 0$  for  $C_{ab}^* = 0$ ) and the gamut-mapped image has  $h_{ab} \simeq -140^\circ$ . Around pixel 192 it is evident that the hue change is  $\Delta h_{ab} \simeq 180^\circ$ .

Even though these two images are built to be symptomatic of different types of drawbacks (and would actually be better treated with spatially invariant GM), they exaggerate problems that can be encountered in normal images too. In the next section we build a framework for multiscale spatial GM in which these drawbacks can be removed or strongly attenuated.

## V. FRAMEWORK FOR MULTISCALE GM

The framework we propose requires an important constraint on the color space in which the gamut mapping is performed: lines of constant hue must be straight lines towards the gray axis of the color space, and the color space must be perceptually homogeneous. A color space that fully accomplishes these two constraints does not exist, and according to some researchers it will also be impossible to construct such a space due to

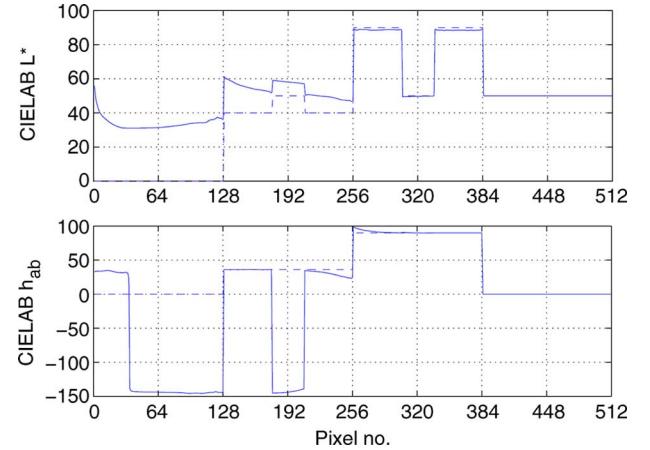


Fig. 4. Scan line plots for a horizontal line going through the center of the  $n_r$  and  $n_g$  patches of the second synthetic image (left hand side of Fig. 3); (top)  $L^*$  and (bottom)  $h_{ab}$ . The solid lines are the scan lines of the reproduced image, and the dashed lines are the scan lines of the original image.

the intrinsic curvature of the color domain, see, e.g., [29]. A well-known color space that approximates these constraints is CIELAB; thus, we use it in this paper as a specific case. However, every color space that approximate these two constraints can be used with the proposed framework.

To solve the above-mentioned drawbacks, mirroring the order presented in Section IV, we now incrementally develop a new framework. First, we force the mapping vectors to lie on *constant hue lines* (Section V-A). Then, we compute the adequate spatially variant compression by means of color properties in the CIELAB color space (Section V-B), trying to preserve local relationship between colors and then using color and spatial properties together to *avoid unwanted compression* (Section V-C). In Section VIII-C, we discuss the treating of  $L^*$ .

### A. Mapping Vectors on Constant Hue Lines

It is evident that the drawbacks of simple multilevel techniques presented in the previous section are induced by wrong direction and magnitude of the mapping vector image  $\Delta f_p = f_{p+1} - f_p$  for some of the colors involved in the mapping process

(cf. Fig. 2). To avoid this problem, we want the mapping vectors of every color to lie on constant hue lines, i.e., straight lines towards the gray axis. Thus, we force the *direction* of the color change  $\Delta\mathbf{f}_p$  to be towards a point on the gray axis, while the *magnitude* of the compression is decided by the underlying GMA. In general, the point on the gray axis can depend on the image itself. Therefore, we introduce the projection  $K : \mathcal{F} \rightarrow \mathcal{F}$  such that  $K(\mathbf{f})(x, y) = k(\mathbf{f}(x, y))\mathbf{e}_L$ , where  $k : \mathbb{R}^3 \rightarrow \mathbb{R}$ , and  $\mathbf{e}_L \in \mathbb{R}^3$  is the unit vector along the gray axis (e.g., the  $L^*$  axis in CIELAB). Then, the following formula describes the approach:

$$\Delta\mathbf{f}_p = s_p(K(\mathbf{f}_p) - \mathbf{f}_p) \Rightarrow \mathbf{f}_{p+1} = \mathbf{f}_p + s_p(K(\mathbf{f}_p) - \mathbf{f}_p) \quad (8)$$

where  $s_p \in \mathbb{R}$  is a scalar. In order to avoid gamut expansion or compression across the gray axis of the color space,  $s_p$  should be restricted to  $s_p \in [0, 1]$ . With this constraint, the image at one stage is a convex linear combination of the image at the previous stage and a gray image, rewriting (8)

$$\mathbf{f}_{p+1} = (1 - s_p)\mathbf{f}_p + s_p K(\mathbf{f}_p). \quad (9)$$

In order to have a spatial variant algorithm,  $s_p$  should be a scalar image  $s_p : \mathcal{I} \rightarrow \mathbb{R}$ . In analogy with the use of the synthetic symbol  $\mathcal{F}$ , we group all such images into the functional space  $\mathcal{S}$ , so  $s_p \in \mathcal{S}$ . Furthermore, in order to have an image dependent algorithm,  $s_p$  should be the result of the application of an operator to the image itself, i.e.,  $s_p = S_p(\mathbf{f}_p)$ , where  $S_p : \mathcal{F} \rightarrow \mathcal{S}$ . In analogy with (1), the operator  $S_p$  could be defined in terms of  $G_{Cl}$  and  $(L_p)$ , but, for notational simplicity, we simply refer to it as  $S_p$ .

More than one spatial operator can be used, since different ones can focus on different properties of the image (e.g., spatial distribution, color, both spatial and color distribution or “higher dimensions”). Thus, a sequence of spatial operators  $(S_p^n)$  can be used simultaneously.

It is, therefore, useful to introduce the  $\phi$  operator for combining the results of  $N$  spatial operators,  $\phi : \mathcal{S}^N \rightarrow \mathcal{S}$ . With this generalization, the complete formulation of the framework for multiscale spatial GM is

$$\mathbf{f}_{p+1} = (1 - \phi)\mathbf{f}_p + \phi K(\mathbf{f}_p) \quad (10)$$

where, for the sake of clarity, with  $\phi$ , we actually mean

$$\phi = \phi \left( (S_p^n(\mathbf{f}_p))_{n=0}^{N-1} \right). \quad (11)$$

Every vector  $\Delta\mathbf{f}_p(x, y)$  is directed towards the gray axis, while the magnitude of the compression is decided by the operators  $(S_p^n)_{n=0}^{N-1}$  through  $\phi$ . The simple multiscale method (1) reduces to conventional spatial invariant GM in the case of trivial operators  $P = 1$ ,  $L_0 = I$ . Unfortunately, it depends upon the choice of the operators  $S_p$  and  $\phi$  whether or not (10) and (11) reduce to conventional spatial invariant GM. In the next section, we show how to design the operator  $S_p^0$  to achieve a magnitude of compression that is consistent with the goals of GM [4].

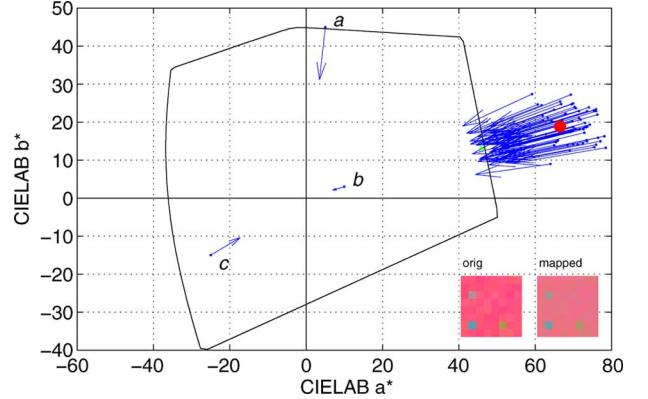


Fig. 5. Resulting color change to the pixels of the image in Fig. 1 when using the proposed  $S_p^0$  operator (12). The synthetic image and the resulting gamut-mapped image are shown in the lower right corner.

### B. Preserving Local Relationships Using Clipping-Driven Compression: Design of $S_p^0$

As a first example on using this framework, we restrict ourselves to one spatial operator  $S_p^0$ , taking  $\phi = I$  (the identity operator), and  $K(\mathbf{f}) = \mathbf{C}$ , where  $\mathbf{C} \in \mathcal{F}$  is the constant image  $\mathbf{C}(x, y) = \mathbf{c}$ , and  $\mathbf{c} \in \mathbb{R}^3$  is the center of the gamut on the gray axis. Like for the simple multiscale scheme (1), we compute the magnitude of compression for every pixel by considering the magnitude of the clipping vector for the low-pass-filtered image,  $|G_{Cl}(L_p(\mathbf{f}_p))(x, y) - L_p(\mathbf{f}_p)(x, y)|$ , where  $|\cdot|$  is the  $L^2$  norm in  $\mathbb{R}^3$ . This amount of compression can not be applied as it is to all pixels, since it may lead to over compression similar to the case of pixel  $b$  in Fig. 2. Instead, a *relative compression* can be applied for all pixels:<sup>2</sup>

$$S_p^0(\mathbf{f}_p) = \frac{|G_{Cl}(L_p(\mathbf{f}_p))(x, y) - L_p(\mathbf{f}_p)(x, y)|}{|\mathbf{C} - L_p(\mathbf{f}_p)|}. \quad (12)$$

Here, the numerator is the amount of compression needed for the average pixel value from the low-pass-filtered image, and the denominator is the maximum possible compression, i.e., the distance from the average color to the center of the color space. The values of  $S_p^0(\mathbf{f}_p)$  are in the range  $[0, 1]$  due to the use of  $G_{Cl}$ .

Applying this method to the example of Fig. 1 (keeping  $(L_p)$  as before), gives the result in Fig. 5. We see that the defined operators  $S_p^0(\mathbf{f}_p)$  and  $\phi = I$  within the framework, simultaneously solve the problems of hue shifts, hue changes and gamut expansion. However, they still introduce some unwanted gamut compression for the pixels  $a$  and  $c$ . Consequently, some haloing will still be present in the resulting image. Running this algorithm on the synthetic image of Fig. 3 gives (left) Fig. 6, and the corresponding scan lines in Fig. 7. It can be noticed that the haloing is still present, but achromatic. The next section presents the design of  $S_p^1(\mathbf{f}_p)$  operator that, combined with  $S_p^0(\mathbf{f}_p)$  and using an appropriate  $\phi$ , reduces the haloing effect.

### C. Avoiding Unwanted Compression: Design of $S_p^1$

The use of  $S_p^0$  guarantees a proper amount of compression for colors that contribute significantly to the mean value; for

<sup>2</sup>Here, we actually introduce a new function  $|\cdot| : \mathcal{F} \rightarrow \mathcal{S}$  defined by the  $L^2$  norm  $|\cdot| : \mathbb{R}^3 \rightarrow \mathbb{R}$ , such that for  $\mathbf{f} \in \mathcal{F}$ ,  $|\mathbf{f}|(x, y) = |\mathbf{f}(x, y)|$ .

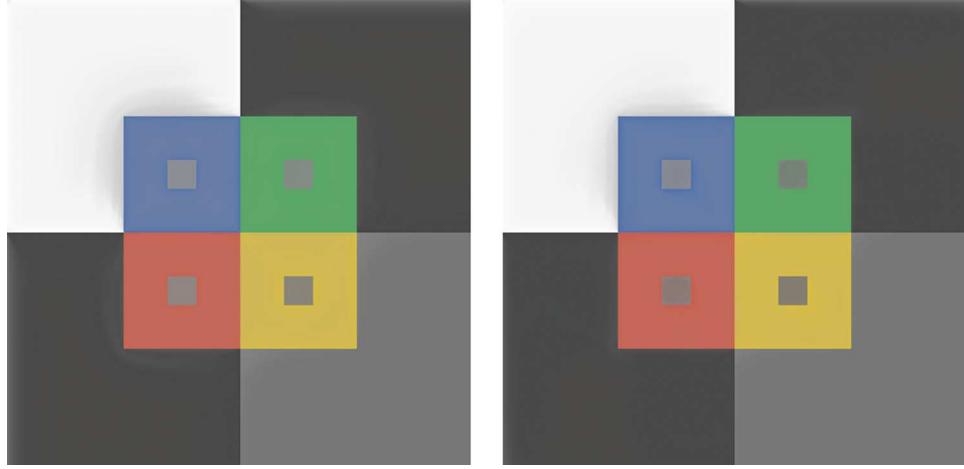


Fig. 6. Synthetic image from Fig. 3 gamut mapped to the ISO uncoated gamut using the proposed framework with the (left)  $S_p^0$  operator and using the (right)  $\phi(S_p^0, S_p^1)$  operators.

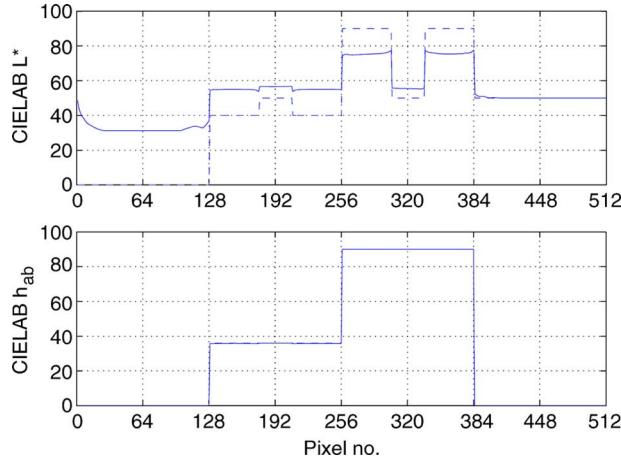


Fig. 7. Same scan lines as in Fig. 4 for the image on the left hand side of Fig. 6.

these colors the direction of mapping is similar to the one of the clipping vector applied to the average color. However, since  $S_p^0$  does not take into account the direction of mapping, the compression magnitudes of pixels  $a$  and  $c$  of Fig. 5 are still significant. Even though the direction of mapping is correct, the compression of  $a$  and  $c$  is somewhat *undesired*. However, for pixel  $b$  in the same figure, some compression is relevant in order to keep good local relationships with respect to similar colors, and avoiding reduced rendering of details. Thus, we need a way to distinguish between colors that should be compressed using the scalar operator  $S_p^0$  and others that should be treated differently.

The angle between the pixel value of the image at level  $p$ ,  $\mathbf{f}_p(x, y)$ , and the corresponding low-pass-filtered one,  $L_p(\mathbf{f}_p(x, y))$ , taken from the center of the gamut  $\mathbf{c}$ , can be used to influence the magnitude of compression. This angle is given by  $\theta_p : \mathcal{F} \rightarrow \mathcal{S}$

$$\theta_p(\mathbf{f}_p)(x, y) = \arccos \left( \frac{(\mathbf{f}_p(x, y) - \mathbf{c}) \cdot (L_p(\mathbf{f}_p)(x, y) - \mathbf{c})}{|\mathbf{f}_p(x, y) - \mathbf{c}| |L_p(\mathbf{f}_p)(x, y) - \mathbf{c}|} \right) \quad (13)$$

where the dot denotes the Euclidean inner product in  $\mathbb{R}^3$ .

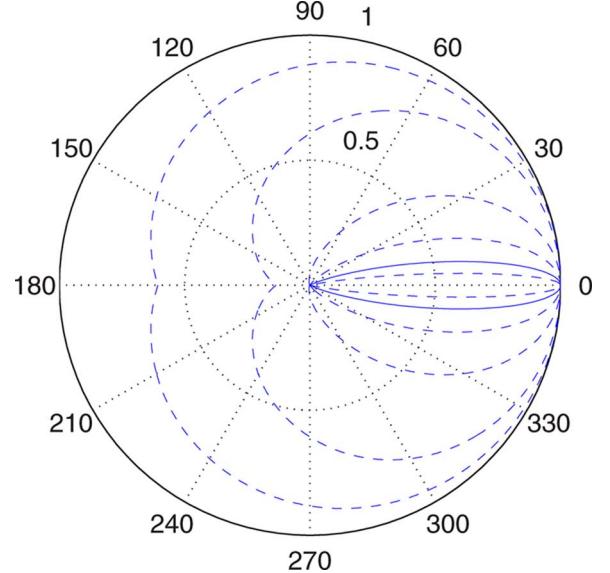


Fig. 8. Polar plot of the relevance operator  $R$  taken as a function of the angle  $\theta$  for the following values of  $\sigma_R$ :  $\pi/40, \pi/20, \pi/10, \pi/5, \pi/2$ , and  $\pi$ . The curve for  $\sigma_R = \pi/20$  is drawn as a solid line.

To link the angular value to the compression magnitude, we build a smooth *relevance operator*  $R_p$ . It should be highest (e.g., equal to 1) for  $\theta = 0^\circ$ , and decrease towards zero when the absolute value of the angle increases. This can be obtained using the Gaussian function; thus, the relevance operator  $R_p : \mathcal{F} \rightarrow \mathcal{S}$ , is defined by

$$R_p(\mathbf{f}_p)(x, y) = \exp \left( -\frac{(\theta_p(\mathbf{f}_p)(x, y))^2}{2\sigma_R^2} \right). \quad (14)$$

While the relevance is always equal to 1 for  $\theta = 0^\circ$ ,  $\sigma_R$  should be tuned to give zero relevance at the appropriate “cutoff” angle. To our knowledge, no theoretical assumptions can indicate the optimal “cutoff” angle. Fig. 8 shows polar plots for different shapes of the relevance operator varying  $\sigma_R$ .

Preliminary tests showed that  $\sigma_R = \pi/20$  (solid line in Fig. 8) is consistent with the aim of  $S_p^1$  operator, and gives the ex-

pected desired benefits to the GMA. It is important to discuss its meaning and the behavior of the GMA varying  $\sigma_R$ . For the sake of completeness and clarity, Section VI-A has been devoted to the discussion of this parameter, avoiding a distracting discussion on  $\sigma_R$  here.

The relevance operator  $R_p$  could be directly used as  $S_p^1$ . However, since  $\theta_p$  is directly dependent on  $\mathbf{f}_p$ ,  $R_p(\mathbf{f}_p)$  will have high-frequency content. This implies that some high-frequency information is injected from higher to lower scales, thus infringing on the principles of the scale-space theory. It is, therefore, necessary, and reasonable,<sup>3</sup> to perform a low-pass filter on  $R_p(\mathbf{f}_p)$  before using it into the framework, thus defining  $S_p^1 : \mathcal{F} \rightarrow \mathcal{S}$  by

$$S_p^1(\mathbf{f}_p) = L_p(R_p(\mathbf{f}_p)). \quad (15)$$

#### D. Combining $S_p^0$ and $S_p^1$ : Design of $\phi$

We defined the two operators  $S_p^0$  and  $S_p^1$ , the first giving us the magnitude of the GM, and the second giving us the relevance of the magnitude prompted by the first one. A reasonable way to compose these two operators is by multiplication, thus devising the  $\phi$  operator as follows:

$$\phi(S_p^0(\mathbf{f}_p), S_p^1(\mathbf{f}_p)) = S_p^0(\mathbf{f}_p)S_p^1(\mathbf{f}_p). \quad (16)$$

With this choice, we get that  $\phi(S_p^0(\mathbf{f}_p), S_p^1(\mathbf{f}_p)) \leq S_p^0(\mathbf{f}_p)$  pixelwise, so  $\phi$  can be seen as a conservative proposal, since the magnitude of GM (i.e.,  $S_p^0$ ) is always decreased (or at least left unchanged).

This is *not* the only way to combine different  $S_p$  operators. As an example, an alternative combination of  $S_p^0$  and  $S_p^1$  can be designed as  $\phi(S_p^0(\mathbf{f}_p), S_p^1(\mathbf{f}_p)) = 1 - (1 - S_p^0(\mathbf{f}_p))(1 - S_p^1(\mathbf{f}_p))$ . This choice can be considered more “aggressive” since  $\phi(S_p^0(\mathbf{f}_p), S_p^1(\mathbf{f}_p)) \geq S_p^0(\mathbf{f}_p)$  pixelwise. This example clarifies the importance of  $\phi$  operator and the modularity of the framework. When the  $\phi$  operator (16) is chosen, this method reduces to conventional spatial invariant GM for the trivial case of one level,  $L_0 = I$ .

From now on, when using the term *proposed* GMA, we refer to the one based on the framework (10), with  $(S_p^n)_{n=0}^{N-1}$  operators ( $N = 2$ ) given by (12) and (15) with  $\sigma_R = \pi/20$  and the  $\phi$  operator (16). The proposed GMA is summarized in Table II.

The application of the proposed GMA to the example of Fig. 1 gives the result in Fig. 9. We see that the problem of unwanted gamut compression for the pixels  $a$  and  $c$  has been reduced thanks to  $S_p^1(\mathbf{f}_p)$ . The result for running this algorithm on the synthetic image of Fig. 3 is shown in (right) Fig. 6, and the corresponding scan lines are shown in Fig. 10.

## VI. DISCUSSION OF THE PROPOSED METHOD

This section discusses the only parameter of the proposed algorithm,  $(\sigma_R)$ , the computational complexity of the algorithm and the drawbacks inherited from spatially invariant GMAs.

<sup>3</sup>Preliminary tests show that high-frequency re-injection results in ringing effect.

TABLE II  
SUMMARY OF THE PROPOSED GMA

$$\mathbf{f}_{p+1} = (1 - S_p^0(\mathbf{f}_p)S_p^1(\mathbf{f}_p))\mathbf{f}_p + S_p^0(\mathbf{f}_p)S_p^1(\mathbf{f}_p)\mathbf{C}$$

$$S_p^0(\mathbf{f}_p) = \frac{|G_{\text{cl}}(L_p(\mathbf{f}_p)) - L_p(\mathbf{f}_p)|}{|\mathbf{C} - L_p(\mathbf{f}_p)|}$$

$$S_p^1(\mathbf{f}_p) = L_p(R_p(\mathbf{f}_p))$$

$$R_p(\mathbf{f}_p)(x, y) = \exp\left(-\frac{(\theta_p(\mathbf{f}_p)(x, y))^2}{2\sigma_R^2}\right)$$

$$\sigma_R = \pi/20$$

$$\theta_p(\mathbf{f}_p)(x, y) = \arccos\left(\frac{(\mathbf{f}_p(x, y) - \mathbf{c}) \cdot (L_p(\mathbf{f}_p)(x, y) - \mathbf{c})}{|\mathbf{f}_p(x, y) - \mathbf{c}| |L_p(\mathbf{f}_p)(x, y) - \mathbf{c}|}\right)$$

$$L_p(\mathbf{f}) = g_{2^{P-2-p}} * \mathbf{f}$$

$$g_{\sigma_p}(x, y) = \frac{1}{2\pi\sigma_p^2} \exp\left(-\frac{x^2 + y^2}{2\sigma_p^2}\right)$$

$$P = \lceil \log_2(\max(W, H)) \rceil - 2$$

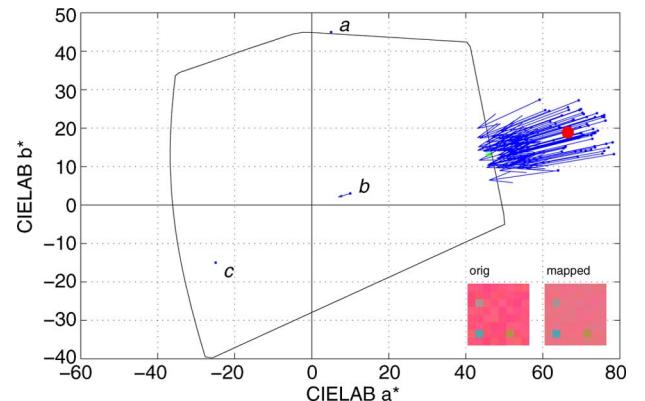


Fig. 9. Resulting color change to the pixels of the image in Fig. 1 when using the proposed  $\phi$  (16),  $S_p^0$  (12), and  $S_p^1$  (15) operators. The synthetic image and the resulting gamut-mapped image are shown in the lower right corner.

#### A. Algorithm Behavior Varying $\sigma_R$

The design of the relevance operator (14) and the choice of  $\sigma_R$  can affect the algorithm behavior. Moreover, the Gaussian shape of the relevance operator could be substituted by alternative smooth functions. Alternatives must be non-negative monotonically decreasing functions of  $|\theta|$  giving 1 for  $\theta = 0^\circ$ .

Regarding the tuning of  $\sigma_R$ , it can be noticed that  $S_p^1(\mathbf{f}_p)(x, y) \rightarrow 1$  when  $\sigma_R \rightarrow \infty$ , thus reducing the algorithm to the one proposed in Section V-B (i.e., without the  $S_p^1$  and with  $\phi = I$  operators). When  $\sigma_R \rightarrow 0$ , and the low-pass filtering is omitted [see (15)], the algorithm turns into

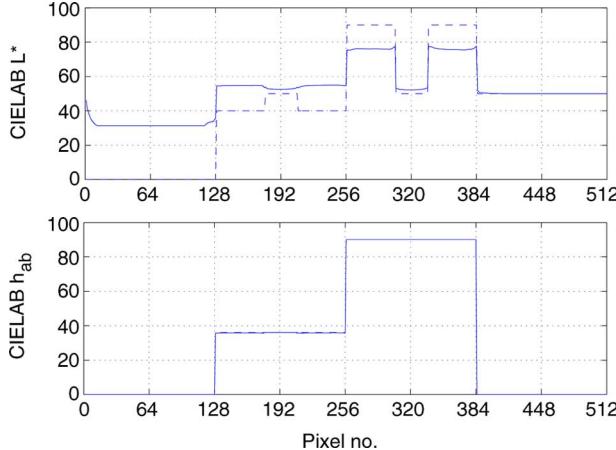


Fig. 10. Same scan lines as in Fig. 4 for the image on the right hand side of Fig. 6.

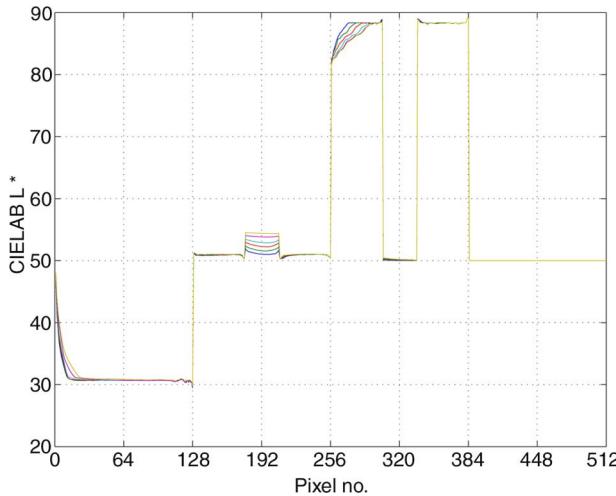


Fig. 11. Same scan lines as in Fig. 4 for the image on the right hand side of Fig. 6 for the following values of  $\sigma_R$ :  $\pi/20, \pi/40, \pi/20, \pi/10, \pi/5, \pi/2$ , and  $\pi$ .

conventional spatial invariant GM, regardless of the choice of  $P$  and  $(L_p)$  (only colors that produce  $\theta = 0$  are compressed). In this case, the use of the low-pass filter  $L_p$  (15) inhibits that the algorithm becomes spatially invariant.

Summarizing,  $\sigma_R \in (0, \infty)$  controls the balance between a clipping-driven global compression ( $\sigma_R \rightarrow \infty$ ) and a quasi-spatially-invariant gamut clipping ( $\sigma_R \rightarrow 0$ ). Since  $S_p^1(\mathbf{f}_p)$  has been devised to avoid unwanted compression with the goal of reducing halos, an intermediate value of  $\sigma_R$  should be used.

Fig. 11 shows the behavior of the proposed GMA when varying  $\sigma_R$ . Decreasing the  $\sigma_R$  value reduces the halos.

### B. Computational Cost and Implementation

The proposed method has a computational complexity equal to  $O(N \log^2 N)$  with  $N$  the number of pixels in the input image. In fact, we must perform  $2 \log N^{1/2} = \log N$  discrete Fourier transforms [two for each scale; see (7) for details]. Using the fast Fourier transform, that is  $N \log N$ , we have to perform

$N \log^2 N$  operations. Computing the operators requires  $N$  operations for every scale; thus, the added operations with respect of a standard multiscale GMA, is  $N \log N$ . These facts give that the proposed algorithm is  $O(N \log^2 N)$ ; thus, we do not increase the computational cost with respect to a standard multiscale GMA.

Regarding the implementation, we extensively used different look up tables to implement in a fast way all the spatially invariant gamut mapping involved in the proposed multiscale approach. The line-gamut surface intersection can be quite expensive. Using these LUTs, we reduce heavily the number of operations of this step obtaining reasonable computational times.

### C. Drawbacks Inherited From Spatially Invariant GMAs

In the previous sections, we omitted to discuss closely the mapping of  $L^*$ . The main reason is that mapping of  $L^*$  is commonly treated separately by the majority of works. Problems with mapping of  $L^*$  are usually inherited from the spatially invariant GMAs used in the multilevel approach. To better discuss the problem, in this section, we only consider spatially invariant GMAs.

To treat  $L^*$ , we have mainly three different standard approaches: clipping toward the center of the destination gamut, compression of  $L^*$  independent of  $a^*$  and  $b^*$ , compression toward the center of the destination gamut [or toward  $(L^*, a^*, b^*) = (100, 0, 0)$ ]. All of these three possibilities have advantages and drawbacks. A simple image that can help to highlight drawbacks is a gradient of increasing  $L^*$  (starting from, e.g.,  $L^* = 10$  to 20) while decreasing chroma  $C_{ab}^*$  (starting from, e.g.,  $C_{ab}^* = 60$  to 20).

If the GMA performs clipping toward the center of the destination gamut, usually it inverts the  $L^*$  gradient but keeps the relationships between chroma values. This is unacceptable since  $L^*$  information is more perceptually important than chroma.

If the GMA performs compression of  $L^*$  independent of  $a^*$  and  $b^*$  (followed by clipping toward the gray axis), it guarantees that the  $L^*$  relationships are maintained, but, because of the gamut shape, the chroma relationships can be inverted. Even though this approach is better than the previous one, we can have a loss of fine details due to the compression of  $L^*$ . This is a well-known consequence of compression techniques.

Finally, if the GMA performs a global compression, both  $L^*$  and  $C_{ab}^*$  relationships are preserved, and no inversion of gradients is induced. Unfortunately, it is well known that compression causes an important loss of contrast and loss of fine details. For these reasons, the use of a spatially invariant compression method alone is more a theoretical than a practical solution.

For the sake of completeness, we recall that the SGCK method [30] proposes a technique that tries to preserve  $L^*$  relationship by taking into account both hue and chroma.

Summarizing, the preservation of  $L^*$  relationships requires a tradeoff solution involving at least the treating of chroma  $C_{ab}^*$  and the detail preservation. This tradeoff solution should be searched by using the appropriate spatially invariant GM into a multiscale approach. Section VIII-C suggests some future extensions of the algorithm to improve the treatment of  $L^*$ .



Fig. 12. Original images used in the GM test shown in Fig. 13. The girl an camera images were provided by the photographer Ole Jakob Bøe Skattum, and the cat and ski images by the CIE TC8-03.

## VII. RESULTS ON REAL IMAGES

Fig. 13 shows the result of running the proposed GMA on the real images shown in Fig. 12. Four images with quite different properties have been chosen. The Girl image is a typical portrait with a flat background and not too saturated colors. This kind of images is often quite well treated by spatially invariant GMAs, whereas spatial algorithms can have problems with the edge between the hair and the background, resulting in halos. The Camera image contains a lot of details in the dark regions that are often strongly attenuated or lost with spatially invariant GMAs, and it is, thus, an example of an image that should be well off with spatially variant GM. The Ski image is also very detailed, and has a lot of information in the chromaticity dimension. We added this image following the guidelines of the CIE [30]. The Cat image is a typical computer graphics image that has strong saturated colors in almost flat areas. Note that there are some halos present in the original. Like the Girl image, this kind of images is often well treated by spatially invariant GMAs.

The pure multilevel scheme (1) gives the strongest rendering of details for all of the images at the cost of introducing strong halos and hue shifts. This is particularly visible in the Girl image, see, e.g., the green halo between hair and neck, and the light halo around the head in the closeups shown in Fig. 14. These halos are strongly attenuated by the proposed algorithm, while still giving a good rendering of detail. The effects of over compression and hue shifts caused by the simple multilevel scheme is particularly evident in the red badge in the Camera image, shown in the closeup in Fig. 15. The hue shift is completely removed by the proposed algorithm, and the over-compression is strongly attenuated.

## VIII. FURTHER EXTENSIONS WITHIN THE PROPOSED FRAMEWORK

The proposed framework exhibits a good modularity. This section shows some further extensions to the proposed GMA, providing also some citations of possible previous research that can be fused into the framework.

### A. Other Color Spaces

As stated in Section V, our framework requires that the color space used in the gamut processing should have some important properties. The CIELAB color space only partially fulfills the requirements. The blue region of CIELAB color space is particularly inhomogeneous [31]. Other color spaces, e.g., the Munsell color space, can perform better than CIELAB with respect to *perceptual uniformity* and *isotropy* and have been suggested as an alternative [32]. The Coloroid System has been proposed for a spatially invariant method by Neumann and Neumann [33] taking advantages of the fact that, quoting their article, “the constant hue planes of the Coloroid system are perceptually perfectly uniform.” However, many other color spaces can be used within the proposed framework; some examples being, e.g., OSA, UCS,  $L_{jg}$ , and CIECAM02.

### B. Other Scale Spaces

We have chosen to use Gaussian kernels to construct the scale-space representation of the image. This is a robust and not too computationally expensive way to obtain the scale space. However, the problem of haloing that has been partly solved by the application of the  $S_p^1$  operator mainly results from this choice.

Scale-space representations that would reduce the haloing problem could be obtained using nonlinear methods such as bilateral filtering [15] or anisotropic diffusion [34]. Confirming this direction of development, the former method has already been implemented by Zolliker [14], reducing the haloing effect. With such scale spaces, there might not be the need of an  $S_p^1$  operator, at the cost of a higher computational complexity.

However, it must be taken into account that the operators  $S_p^0$  and, in particular,  $S_p^1$  have been specifically designed for the Gaussian scale-space representation. For other types of scale spaces to be successfully applied in the framework, the  $S$  operators must be redesigned accordingly.

### C. Design of $G$ and $K(\mathbf{f})$ for Preserving $L^*$ Relationship

Even though the selection and/or design of  $G$  and  $K(\mathbf{f})$  is not the main topic of the paper, we already implicitly defined

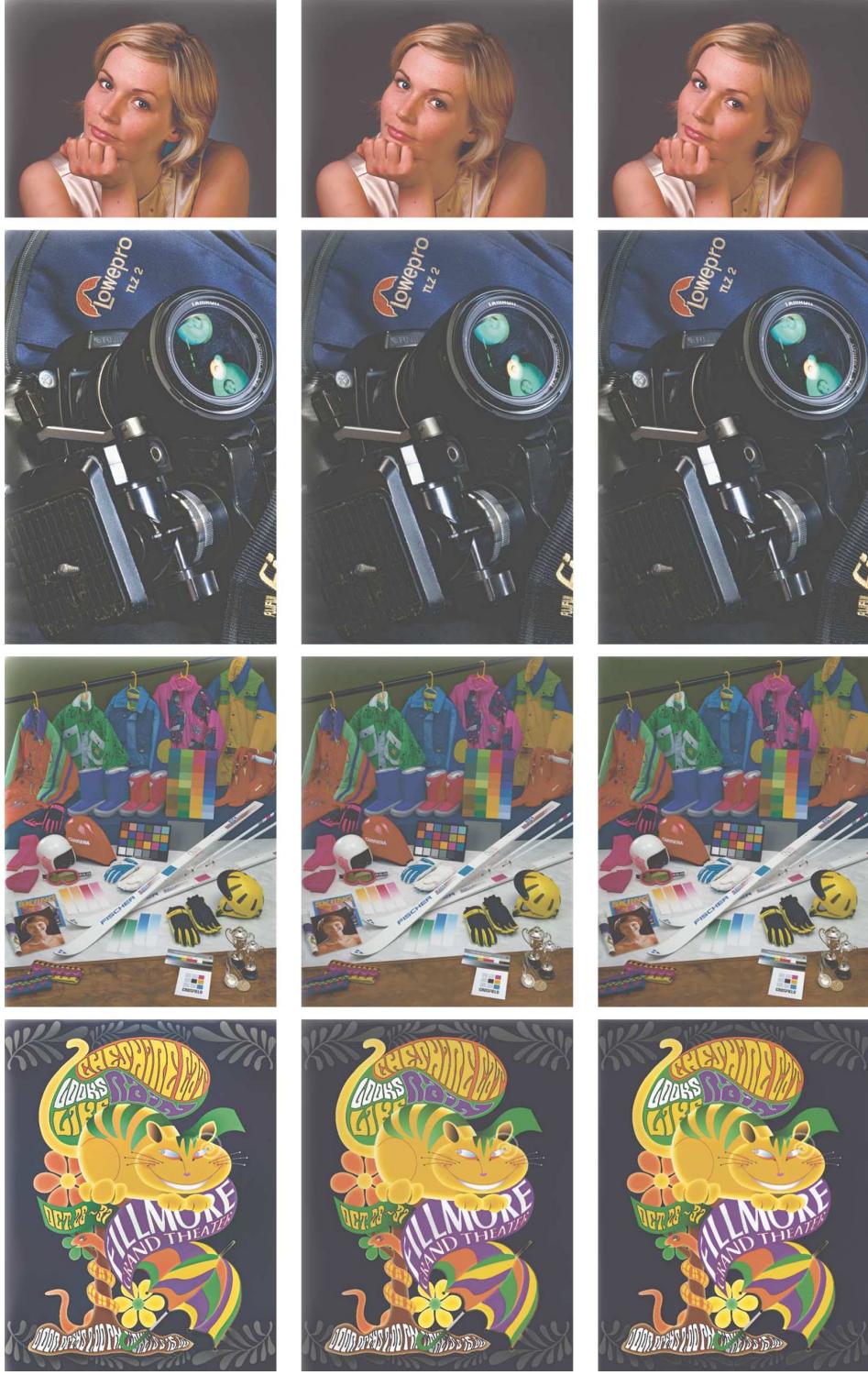


Fig. 13. Results running different GMAs on real images from Fig. 12 for gamut mapping to the ISO uncoated gamut. The first column presents the results of the simple multilevel scheme (1), the second presents the results using the proposed framework (10) with the  $S_p^0$  operator (12), the third presents the results of the proposed algorithm using the  $\phi$  (16),  $S_p^0$  (12), and  $S_p^1$  (15) operators.

the pair  $(G_{\text{cl}}, \mathbf{C})$  in the previous sections. We briefly refer to this approach as *toward center*.

For the sake of completeness and to show the generality of the proposed framework, we now add two new different pairs  $(G, K(\mathbf{f}))$  with the goal to preserve the  $L^*$  relationships while trying to avoid the main drawbacks discussed in Section VI-C.

*1) Sequential Mapping:* The first proposal takes inspiration from a standard way to treat  $L^*$ . We first gamut map the lightness and then the chromaticity. A lightness image  $g_0 = A(\mathbf{f}_0)$  is obtained by the projection  $A : \mathcal{F} \rightarrow \mathcal{F}$ ,  $A(\mathbf{f})(x, y) = f_{L^*}(x, y)\mathbf{e}_{L^*}$ ,  $f_{L^*}(x, y)$  being the lightness component of  $\mathbf{f}(x, y)$ . Thus, the mapping of lightness (with



Fig. 14. Detail of the girl image; original, simple multilevel scheme (1) and the proposed algorithm of Table II.



Fig. 15. Detail of the camera image; original, simple multilevel scheme (1) and the proposed algorithm of Table II.

$P$  the number of scales) is obtained using (10) on the image  $g_0$  using the pair  $(G_{cl}, \mathbf{C})$ . Denoting by  $\mathbf{g}_P$  the result of the multiscale lightness mapping, we substitute the original lightness with the gamut-mapped one obtaining  $\tilde{\mathbf{f}}_0 = \mathbf{f}_0 - g_0 + \mathbf{g}_P$ . This full color, lightness compressed image, is then used as input image for the proposed multiscale GMA using the pair  $(G_{cl,A(\mathbf{f})}, A(\mathbf{f}))$ , i.e., mapping towards the gray axis along lines of constant hue and lightness.

2) *Toward Compressed Gray Axis:* We also propose a novel alternative method for lightness mapping, by incorporating it into a unique multiscale computation, without the necessity of a sequential mapping. The main idea is to compute the direction of mapping according to  $g_0$ . In this case, it is possible to map the lightness toward the compressed lightness on the gray axis. Denoting by  $M_g$  and  $m_g$  the maximum and minimum lightness on the gray axis of the destination gamut respectively, this can be achieved by the projection  $K : \mathcal{F} \rightarrow \mathcal{F}$ ,  $K(\mathbf{f})(x, y) = (m_g + f_{L^*}(x, y)(M_g - m_g)/100)\mathbf{e}_{L^*}$ . With this projection, the suggested approach is implemented using the pair  $(G_{cl,K(\mathbf{f})}, K(\mathbf{f}))$ .

## IX. CONCLUSION

We presented a unified viewpoint for analyzing a class of the latest generation multiscale spatially variant gamut mapping algorithms. This viewpoint encompass at least two of the more important gamut mapping algorithms. Analyzing the framework, we highlighted advantages and drawbacks of multiscale algorithms. We showed the main theoretical and practical causes of these drawbacks and then we built up a novel multiscale framework for spatial gamut mapping. Within this framework, we designed specific operators and functions to solve or strongly reduce these drawbacks. The computational complexity of the proposed gamut mapping algorithm is the same as of a standard multiscale approach. Results on both paradigmatic challenging synthetic images and on real images are presented and show that

the proposed framework can be a robust starting point for new spatially variant gamut mapping algorithms.

## ACKNOWLEDGMENT

The authors would like to thank Dr. E. Provenzi, M. Fierro, Dr. F. A. Cheikh, Ø. Kolås, and A. M. Bakke for useful suggestions and discussions. They would also like to thank the reviewers for helping them improve the paper.

## REFERENCES

- [1] R. W. G. Hunt, *The Reproduction of Colour in Photography, Printing & Television*, 5th ed. London, U.K.: Fountain, 1995.
- [2] I. C. Concoritum, Specification ICC.1:2001-12. File Format for Color Profiles (Ver. 4.0.0) 2001.
- [3] R. Eschbach, "Image reproduction: An oxymoron?", in *Proc. IS&T and SID's 11th Color Imaging Conf.: Color Science and Engineering: Systems, Technologies, Applications*, Scottsdale, AZ, 2003, pp. 2–5.
- [4] J. Morović and M. R. Luo, "The fundamentals of gamut mapping: A survey," *J. Imag. Sci. Technol.*, vol. 45, no. 3, pp. 283–290, 2001.
- [5] R. Bala, "Challenges in color reproduction: Towards higher dimensions," in *Proc. Color Imaging X: Processing, Hardcopy, and Applications*, San Jose, CA, 2005, pp. 162–169.
- [6] J. Meyer and B. Barth, "Color gamut matching for hard copy," in *Proc. SID Dig.*, 1989, pp. 86–89.
- [7] R. Balasubramanian, R. deQueiroz, R. Eschbach, and W. Wu, "Gamut mapping to preserve spatial luminance variations," in *Proc. IS&T and SID's 8th Color Imaging Conf.: Color Science and Engineering*, Scottsdale, AZ, 2000, pp. 122–128.
- [8] R. Bala, R. deQueiroz, R. Eschbach, and W. Wu, "Gamut mapping to preserve spatial luminance variations," *J. Imag. Sci. Technol.*, vol. 45, no. 5, pp. 436–443, Sep./Oct. 2001.
- [9] R. Eschbach, R. Bala, and R. deQueiroz, "Simple spatial processing for color mappings," *J. Electron. Imag.*, vol. 13, no. 1, pp. 120–125, January 2004.
- [10] J. Morović and Y. Wang, "A multi-resolution, full-colour spatial gamut mapping algorithm," in *Proc. IS&T and SID's 11th Color Imaging Conf.: Color Science and Engineering: Systems, Technologies, Applications*, Scottsdale, AZ, 2003, pp. 282–287.
- [11] J. J. McCann, "A spatial colour gamut calculation to optimise colour appearance," in *Colour Image Science*, L. W. MacDonald and M. R. Luo, Eds. New York: Wiley, 2002, pp. 213–233.
- [12] E. H. Land and J. J. McCann, "Lightness and retinex theory," *J. Opt. Soc. Amer. A*, vol. 61, no. 1, pp. 1–11, Jan. 1971.

- [13] E. H. Land, "The retinex theory of color vision," *Sci. Amer.*, vol. 237, pp. 108–128, 1977.
- [14] P. Zolliger and K. Simon, "Retaining local image information in gamut mapping algorithms," *IEEE Trans. Image Process.*, vol. 16, no. 3, pp. 664–672, Mar. 2007.
- [15] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *Proc. 6th Int. Conf. Computer Vision*, Washington, DC, 1998, pp. 839–846.
- [16] S. Nakuchi, M. Immura, and S. Usui, "Color gamut mapping by optimizing perceptual image quality," in *Proc. IS&T and SID's 4th Color Imaging Conf.: Color Science, Systems and Applications*, Scottsdale, AZ, 1996, pp. 63–66.
- [17] R. Kimmel, D. Shaked, M. Elad, and I. Sobel, "Space-dependent color gamut mapping: A variational approach," *IEEE Trans. Image Process.*, vol. 14, no. 6, pp. 796–803, Jun. 2005.
- [18] R. Kimmel, M. Elad, D. Shaked, R. Keshet, and I. Sobel, "A variational framework for retinex," *Int. J. Comp. Vis.*, vol. 52, no. 1, pp. 7–23, 2003.
- [19] P. J. Burt, "Fast filter transforms for image processing," *Comput. Vis., Graph., Image Process.*, vol. 16, pp. 20–51, 1981.
- [20] J. L. Crowley, "A representation for visual information," Ph.D. dissertation, Robotics Inst., Carnegie-Mellon Univ., Pittsburgh, PA, 1981.
- [21] L. Williams, "Pyramidal parametrics," in *ACM SIGGRAPH Computer Graphics*. New York: ACM, 1983, vol. 17, pp. 1–11, no. 3.
- [22] T. Lindeberg, "Scale-space theory: A basic tool for analysing structures at different scales," *J. Appl. Statist.*, vol. 21, no. 2, pp. 224–270, 1994.
- [23] J. J. McCann, "Lessons learned from mondrians applied to real images and color gamuts," in *Proc. IS&T and SID's 7th Color Imaging Conf.: Color Science, Systems and Applications*, Scottsdale, AZ, 1999, pp. 1–8.
- [24] *Colorimetry—Part 4: Cie 1976 1 \* a \* b\* Colour Spaces*, CIE Draft Standard DS 014-4.2/E:2006, CIE, 2006.
- [25] R. Balasubramanian and E. Dalal, "A method for quantifying the color gamut of an output device," presented at the SPIE Color Imaging: Device-Independent Color, Color Hard Copy, and Graphic Arts II, Jan. 1997.
- [26] A. M. Bakke, J. Y. Hardeberg, and I. Farup, "Evaluation of gamut boundary descriptors," in *Proc. IS&T and SID's 14th Color Imaging Conf.: Color Science and Engineering: Systems, Technologies, Applications*, Scottsdale, AZ, 2006, pp. 50–55.
- [27] R. S. Eaton, M. R. Stevens, J. C. McBride, G. T. Foil, and M. S. Snorrason, "A systems view of scale space," presented at the 4th IEEE Int. Conf. Computer Vision Systems, 2006.
- [28] *Graphic Technology—Process Control for the Production of Half-Tone Colour Separation, Proof and Production Prints*, ISO 12647-2:2004, 2004.
- [29] G. Wyszecki and W. S. Stiles, *Color Science—Concepts and Methods, Quantitative Data and Formulae*. New York: Wiley, 1982.
- [30] Guidelines for the evaluation of gamut mapping algorithms, Tech. Rep. 156 CIE, 2003, C. T. C. 8-03.
- [31] N. Moroney, "A hypothesis regarding the poor blue constancy of cielab," *Color Res. Appl.*, vol. 28, no. 5, pp. 371–378, Oct. 2003.
- [32] J. J. McCann, "Color spaces for color-gamut mapping," *J. Electron. Imag.*, vol. 8, no. 4, pp. 354–364, 1999.
- [33] L. Neumann and A. Neumann, "Gamut clipping and mapping based on the coloroid system," in *Proc. CGIV 2nd Eur. Conf. Color in Graphics, Imaging, and Vision*, Apr. 2004, pp. 548–555.
- [34] P. Perona and J. Malik, "Scale-space and edge detection using anisotropic diffusion," *IEEE Trans. Image Process.*, vol. 12, no. 7, pp. 629–639, Jul. 1990.



**Ivar Farup** received the M.Sc. degree in physics from the Norwegian University of Science and Technology, Trondheim, Norway, in 1994, and the Ph.D. degree in applied mathematics from the University of Oslo, Oslo, Norway, in 2000.

Since 2000, he has been an Associate Professor at Gjøvik University College, Gjøvik, Norway, mainly focusing on color imaging.



**Carlo Gatta** received the degree in electronic engineering from the University of Brescia, Brescia, Italy, and the Ph.D. degree from the University of Milan, Milano, Italy.

Since 2001, he has been doing research in the field of digital imaging and human vision.



**Alessandro Rizzi** received the degree in computer science from the University of Milano, Milano, Italy, and the Ph.D. degree in information engineering from the University of Brescia, Brescia, Italy.

He taught information systems and computer graphics at the University of Brescia and at the Politecnico di Milano, Milano, Italy. He is currently an Assistant Professor at University of Milano teaching multimedia and human-computer interaction. Since 1990, he has been doing research in the field of digital imaging and vision. His main research topic is the use of color information with particular attention to color adaptation mechanisms.