
Mila Nikolova (Senior Member IEEE), and Gabriele Steidl

Abstract—Color image enhancement is a complex and challenging task in digital imaging with abundant applications. Preserving the hue of the input image is crucial in a wide range of situations. We propose simple image enhancement algorithms which conserve the hue and preserve the range (gamut) of the R, G, B channels in an optimal way. In our setup, the intensity input image is transformed into a target intensity image whose histogram matches a specified, well-behaved histogram. We derive a new color assignment methodology where the resulting enhanced image fits the target intensity image. We analyse the obtained algorithms in terms of chromaticity improvement and compare them with the unique and quite popular histogram based hue and range preserving algorithm of Naik and Murthy. Numerical tests confirm our theoretical results and show that our algorithms perform much better than the Naik-Murthy algorithm. In spite of their simplicity, they compete with well-established alternative methods for images where hue-preservation is desired.

Index Terms—color image enhancement, scaling and shifting methods, hue preservation, gamut problem, exact histogram specification, color perception.

I. INTRODUCTION

This paper assists to the tremendous progress in digital color imaging and display technology. In spite of the important amount of research, color perception and color appearance are still open problems. The demand for fast efficient algorithms improving the color content of digital images has increased dramatically. The applications of color image improvement are abundant. They concern for example digital cameras and mobile phone cameras, medical imaging, video, post-production industry, restoration of old pictures and movies.

Typically, color images are stored and viewed using three components (channels): red (R), green (G) and blue (B). In this paper we aim to design color image enhancement methods in the RGB space sharing three important features, namely hue and range (gamut) preservation and low computational complexity. The hue describes in each area of an image the dominant color ingredient that one really perceives, e.g., red, orange, magenta, yellow and so on [1], [2]. The hue has the nice property of being invariant under changes of direction and intensity of the incident light [3]. Thus, by preserving the hue and enhancing the brightness, the obtained image will appear more colorful. Examples where the hue is modified are shown in Fig. 1. The range (gamut) preservation is often omitted in works on image enhancement; see, e.g. the recent textbook [2, p. 80]. Each color channel in a digital image can only take a limited number, say $L$, of integer values, e.g., $L = 256$ for 8-bit coding. If the enhancement method produces larger or smaller values these are clipped back to the boundary of $[0, L - 1]$ which also changes the hue. In Fig. 1 right 36.1 % of the pixels are clipped back to 255 which yields too many yellow pixels. Finally, a low computational complexity of algorithms is particularly important when dealing with “mega-pixel” images taken by commercial cameras, resources in hardware implementations and extensions to video.

Remark 1. Fully automatic color image enhancement faces (at least) two major limits: i) “The chemical compounds that form color receptors vary among the population. The physical shapes of the receptors vary among the population and within the retina. Thus, the color vision among observers with normal color vision varies significantly.” [1, p. 18]. ii) Image enhancement is always driven by an application: typically the user needs specific visual information determined by his/her purpose. Further subjective criteria are of paramount importance [2].

Consequently, we do not look for fully automatic image enhancement algorithms. Here we focus on histogram based methods. The selection of a suitable target histogram enables the user’s needs to be satisfied. Moreover we wish to conceive fast algorithms. In order to achieve our goals, we propose simple algorithms composed of two stages:
(a) the intensity channel of the input RGB image is matched to a specified histogram which gives us the target intensity image;
(b) the RGB color values are computed based on the target
intensity image so that they satisfy the hue and gamut constraints in an optimal way. These stages are briefly commented below.

Stage (a). Exact histogram specification (HS), also known as histogram matching, of single-valued (gray-valued) images aims to transform an input image to an output image which exactly fits a prescribed target histogram. Histogram equalization (HE) is a particular case of HS where the target histogram is uniform. Usually HE leads to unnatural images and should not be the target of choice. We do not focus on the construction of target histograms. Instead, we adopt a simple approach inspired by [4]. For digital image HS is an ill posed problem [5]. The clue to ensuring exact HS is to obtain a meaningful total strict ordering of all pixels in the input digital image. We perform exact HS using the algorithm in [32] which currently provides the best pixel ordering in terms of quality and speed.

Stage (b). The extension of histogram methods to color images is a quite complex task. The histogram of a gray-value image is 1-D while the histogram of a color image is 3-D which gives rise to an under-determined problem. For instance, applying HE to each color channel independently changes the color content (the hue) of the image, see Fig. 1 middle. Further it is not easy to produce color images that respect the range constraints; see Fig. 1 right. As a central result of this paper, we propose a general and optimal hue and range preserving color assignment methodology.

Related work. Since the inaugural paper [6] providing PDE-based and variational formulations for image histogram modifications, these methods were further expanded to deal with color image enhancement; see, e.g., [7], [8], [9], [10]. These approaches provide flexible tools to incorporate various knowledge on human visual perceptual phenomena, typically in relation with Retinex theory [11]. An automatic color enhancement (ACE) algorithm for digital images, mimicking some characteristics of the human visual system, has been proposed in [12] and refined in [13]. A fast implementation of ACE was developed in [14]. A perceptually inspired variational approach allowing a more flexible control of contrast adjustment and attachment to data was proposed in [7]. A numerical implementation of the gradient descent technique applied to the corresponding energy functionals coincides with the equation of the ACE. Some basic requirements for "perceptually inspired" objectives were formulated in [8] and gave rise to successful algorithms [8], [15].

Next we summarize the main approaches via histogram modification of color images following a chronological order. Since the suitably normalized histogram of an image is also the empirical probability distribution of its pixel values, a statistical vocabulary is used in many papers. In [16] a 3-D color histogram in the RGB color space was proposed for HE; the resultant images present an excessive brightness for bright pixels, see [17]. A method that preserves both the hue and the range (gamut) constraints was inaugurated by Naik and Murthy in [18]. Even though this article did not show color image applications, it is a state-of-the-art method applied in many papers; see, e.g., [19], [17]. As to the choice of the color space, some methods work directly in the RGB space while others operate in transformed color spaces, e.g., LHS, HSI, YIQ, HSV, etc., see [5]. When processing is done in a transform color space, coming back to the original RGB space typically generates a gamut problem, as cautioned in [18]. Beyond the additional numerical cost, a post-processing in RGB is then needed (often realized using [18]). Gray-value grouping was tentatively extended to color HE in [20]. In [21], a new definition of the histogram of a color image was introduced whose cumulative distribution function (cdf) is the product of the marginal cdf’s of each color channel. Then the color values are increased / decreased by the same amount iteratively. This work was refined in a later paper [22]. Another approach, developed in [19], is to work in the HSI space where the hue and the saturation are equalized and then processed using probability smoothing. All pixels in the RGB space that present gamut problem are corrected using [18]. A generic brightness preserving dynamic histogram equalization scheme, composed of five steps, was proposed in [23]. This scheme was applied to color images in several ways, including transforms into other color spaces. The work in [17] demonstrates that the methods in [19], [21], [16] based on higher dimensional histogram definition, increase the brightness of the image and cannot fit the prescribed uniform histogram. The main conclusion is that only the 1-D histogram of the intensity channel can be considered for equalization. The new color values are then computed using the algorithm in [18]. The method in [17] was recently improved in [24]. In order to avoid the excessive contrast enhancement due to HE, a histogram mixing strategy was applied in [25]. There are also many histogram based techniques where the enhancement function is an S-type, or power, or logarithmic transform; see, e.g., [26], [27], [28], [29]. In particular, the approach in [27] is based on models for color perception and is automatic. Unfortunately, there are no algorithms nor tests on color images.

Contributions. We propose a general affine model for fast hue and range preserving image enhancement in the RGB space which gives rise to Algorithm 3. Two simple but important instances of this algorithm are the Multiplicative algorithm 4 and the Additive algorithm 5. We show how the outcome of Algorithm 3 can be faithfully approximated as a convex combination of the images obtained by Algorithm 4 and Algorithm 5, which is quite practical. The enhancement performances of our algorithms and the Naik-Murthy algorithm [18] are analyzed in terms of their chromaticity improvement. In all cases, our algorithms clearly outperform the algorithm in [18] recently applied to color images in [17]. All numerical tests confirm our theoretical results. Our algorithms are simple and fast. They are really efficient when one wishes to give a better clarity of images (not too altered by artifacts) while preserving the original color ambiance.

Outline. In Section II we sketch our HS method and present the Naik-Murthy algorithm [18]. Section III presents our approach for color image enhancement. In Section IV we evaluate our algorithms and the algorithm in [18] analytically in terms of saturation as well as qualitatively. Section V presents numerical results. Conclusions and points for future work are drawn in Section VI.

The proofs of all statements are given in the Appendix.
II. Preliminaries

Let \( w = (w_r, w_g, w_b) \) be an RGB image of size \( M \times N \), where \( w_c \in \{0, \ldots, L-1\}, c \in \{r, g, b\} \) are its red, green and blue channels, respectively. For 8-bit images we have \( L = 256 \). We reorder each color channel columnwise into a vector of size \( n := MN \) and address the pixels by the index set \( \mathbb{I}_n := \{1, \ldots, n\} \).

A. Histogram Specification

The intensity of an RGB image \( w \) is defined by [5]

\[
f(w) := \frac{1}{3}(w_r + w_g + w_b).
\]

Then \( f \) has \( 3(L-1)+1 \) different values in \( \frac{1}{3}\{0, \ldots, 3(L-1)\} \).

Remark 2. Instead of the intensity \( f \) we can also work with other convex combinations of RGB values. E.g., we can use the weights 0.299, 0.587 and 0.114 which are in proportion to the human perception of the RGB channels, see [1], [2].

We want to find an intensity image \( \hat{f} \) with gray values in \( \{0, \ldots, L-1\} \) which has a specified (target) histogram \( \hat{h} = (\hat{h}_1, \ldots, \hat{h}_L) \), i.e., \( \hat{h}[k] := \frac{\varepsilon}{n} \{i \in \mathbb{I}_n : f[i] = k-1\} \), \( k = 1, \ldots, L \), where \( \varepsilon \) stands for cardinality. Such exact HS can almost never be achieved for images with a small number of different values compared to the number of pixels using the classical statistical method based on the cumulative density function, see [5]. Instead we will apply a procedure based on a meaningful strict ordering of the pixels in \( f \). Various ordering algorithms for digital images were proposed in the literature, see e.g., [30], [31], [33]. The method in [32], based on [33], provides currently the best way in terms of speed and quality to order the pixels in digital images. The basic idea is to minimize a smoothed \( \ell_1 \)-TV functional by simple fixed point iterations with the original image as initialization. After a few iterations the approximate minimizer has entries which differ (up to very few outliers) pairwise from each other while the ordering of the original gray values is retained. Let \( \nabla \) denote the discrete gradient operator (horizontal and vertical forward differences), see [32], \( \nabla^T \) its transposed and let

\[
\eta(t) := \frac{t}{\alpha + |t|} \quad \text{and} \quad \eta^{-1}(y) = \frac{\alpha y}{1 - |y|},
\]

where \( \alpha := 0.05 \) is the default value. Note that \( \eta = \theta' \) where \( \theta(t) := |t| - \alpha \log(1 + |t|/\alpha) \), see [32]. Once a strict ordering is obtained, exact HS is direct. Our HS algorithm reads as:

Algorithm 1 HS using strict ordering [32]

Initialization: \( u^{(0)} := f \), \( \beta := 0.1 \), target histogram \( \hat{h} \), iteration number \( K \) (default \( K := 5 \)), \( c_0 := 0 \).

1. For \( k = 1, \ldots, K \) compute
   \[
u^{(k)} := f - \eta^{-1}(\beta \nabla^T \eta(\nabla u^{(k-1)})).
\]
2. Obtain the ordering \( \{i_j\}^n_{j=1} \) of \( \mathbb{I}_n \) from the ascending sort of the entries of \( u^{(K)} \).
3. For \( k = 0, \ldots, L-1 \) set \( c_{k+1} := c_k + h_k \) and \( \hat{f}[c_k + 1] = \ldots = \hat{f}[c_{k+1}] = k \).

The importance of a meaningful ordering for HS is illustrated in Fig. 2 in the context of HE. The Matlab built-in function histeq does not involve a strict ordering of \( f \) and the resulting histogram of \( \tilde{f} \) is not uniform. This entails some artifacts shown in (a). Such artifacts are not observed in (b) obtained using our Algorithm 1. The colors in Fig. 2(a)-(b) were assigned using Algorithm 2 given in the next subsection.

B. Hue and Range Preservation

Range preservation is a mandatory constraint for all digital imaging devices [4]. A transformed version \( \tilde{w} \) of \( w \) can be correctly depicted only if

\[
\tilde{w}_c[i] \in [0, L-1] \quad \forall \ i \in \mathbb{I}_n \quad \forall \ c \in \{r, g, b\},
\]

since no more than \( L \) digits can be displayed. Otherwise, the obtained image is modified according to the visualization device - which is quite an ad-hoc option; see e.g., Fig. 1 right.

The hue of an image \( w \) is given by \( H(w) := 0 \) if \( w_r = w_g = w_b \) and otherwise by

\[
H(w) := \begin{cases} \theta & \text{if } w_b \leq w_g, \\ 360^\circ - \theta & \text{if } w_b > w_g, \end{cases}
\]

where

\[
\theta := \arccos \left( \frac{\frac{1}{3}(w_r-w_g) + (w_r-w_b)}{(w_r-w_g)^2 + (w_r-w_b)(w_g-w_b))^{\frac{1}{2}}} \right),
\]

see [5]. Note that the denominator of \( \theta \) can be rewritten as \( (\frac{1}{3}(w_r-w_g)^2 + (w_r-w_b)^2 + (w_g-w_b)^2))^{\frac{1}{2}} \).

Remark 3. The simplest hue and range preserving method is to apply the same affine mapping \( \xi(w) := aw + b \) to all pixels, computing \( a \) and \( b \) so that the least and the largest pixels in \( \xi(w) \) are 0 and \( L-1 \), respectively. Let \( w_{\max} := \max\{w_c[i] ; c \in \{r, g, b\}, i \in \mathbb{I}_n\} \) and let \( w_{\min} := \min\{w_c[i] ; c \in \{r, g, b\}, i \in \mathbb{I}_n\} \). Then \( \xi(w) \) given by

\[
\xi(w) := (L-1) \frac{w - w_{\min}}{w_{\max} - w_{\min}} (b)
\]

is the desired stretching of \( w \). For example, see Fig. 16, top.

It is easy to see that the hue of the modified image \( \tilde{w} \) is also preserved if the color values of each pixel are modified by the same affine transform

\[
\tilde{w}_c[i] = a[i]w_c[i] + b[i], \quad c \in \{r, g, b\},
\]

where the constants \( a[i] \) and \( b[i] \) have to be chosen for any \( i \in \mathbb{I}_n \). Finding other appropriate hue-preserving transforms is
an interesting problem. For \( a[i] = 1 \), model (6) amounts to an 
additive transform, usually called shifting. For \( b[i] = 0 \), it is a 
linear/multiplicative transform known as scaling. Both scaling 
and shifting have been introduced in [34], [35]. In general, the 
result of (6) fails the range constraint (3).

The gamut problem was examined by Naik and Murthy in 
[18] in the scaling case for \( a[i] = \hat{f}[i]/f[i] \), where \( \hat{f} \) is a target 
intensity. If \( \hat{f}[i]/f[i] > 1 \), the range constraint (3) might not 
be guaranteed. In such a case the authors propose to avoid the 
potential problem by switching from the RGB color space to 
the CMY (Cyan = \( L - 1 \)-R, Magenta = \( L - 1 \)-G, Yellow 
= \( L - 1 \)-B) space and then to transform back into RGB. This 
correction step reads for all \( c \in \{r, g, b\} \) as

\[
\hat{w}_c[i] = L - 1 - \frac{L - 1 - \hat{f}[i]}{L - 1 - f[i]} \left(L - 1 - w_c[i]\right) \text{ if } \frac{\hat{f}[i]}{f[i]} > 1.
\]

This formula is equivalent to

\[
\hat{w}_c[i] = \frac{L - 1 - \hat{f}[i]}{L - 1 - f[i]} (w_c[i] - f[i]) + \hat{f}[i],
\]

so that the algorithm in [18] can be formulated as follows:

**Algorithm 2** Naik and Murthy [18]

1. Compute the intensity \( f \) of \( w \) and the target intensity \( \hat{f} \).
2. For \( i \in \mathbb{I}_n \), compute
   
   (i) \( \hat{w}_c[i] := \frac{\hat{f}[i]}{f[i]} w_c[i] \) \text{ if } \frac{\hat{f}[i]}{f[i]} \leq 1
   
   (ii) \( \hat{w}_c[i] := \frac{L - 1 - \hat{f}[i]}{L - 1 - f[i]} (w_c[i] - f[i]) + \hat{f}[i] \) \text{ if } \frac{\hat{f}[i]}{f[i]} > 1

Algorithm 2 is often used to avoid the gamut problem.

### III. NEW AFFINE HISTOGRAM SPECIFICATION MODELS

In this section we develop our affine color enhancement methodology. Given an RGB image \( w \) and a target histogram, we compute its intensity \( f \) by (1) and then the target intensity 
image \( \hat{f} \) by Algorithm 1. Our next goal is to transform \( w \) into 
an image \( \hat{w} \) having the following properties:

(a) Intensity fit: \( \hat{f} = \frac{1}{3} (\hat{w}_r + \hat{w}_g + \hat{w}_b) \).
(b) Hue preservation: the hue of \( \hat{w} \) and \( w \) coincide.
(c) Range preservation: \( 0 \leq \hat{w}_c \leq L - 1 \), \( c \in \{r, g, b\} \).

We adopt the hue preserving affine transform (6). Summing up 
over \( c \) in (6) shows that property (a) holds if and only if

\[
\hat{f}[i] = a[i] f[i] + b[i] \iff b[i] = \frac{\hat{f}[i]}{f[i]} - a[i] f[i].
\]

Therefore the affine model (6) obeys (a) if and only if

\[
\hat{w}_c[i] = a[i] (w_c[i] - f[i]) + \hat{f}[i], \quad c \in \{r, g, b\}.
\]

(8)

Two particular instances of (6) are the following:

- Scaling: For \( b[i] = 0 \), model (8) reads as

\[
\hat{w}_c[i] = \frac{\hat{f}[i]}{f[i]} w_c[i], \quad c \in \{r, g, b\}.
\]

(9)

- Shifting: For \( a[i] = 1 \), model (8) becomes

\[
\hat{w}_c[i] = w_c[i] - f[i] + \hat{f}[i], \quad c \in \{r, g, b\}.
\]

We have to adapt these models so that they preserve the range. 
We will use for all \( i \in \mathbb{I}_n \) the magnitudes

\[
M[i] := \max\{w_c[i] : c \in \{r, g, b\}\},
\]

\[
m[i] := \min\{w_c[i] : c \in \{r, g, b\}\}
\]

and similarly \( \hat{M}[i] \) for the maximum and \( \hat{m}[i] \) for the 
minimum of the RGB components of \( \hat{w}[i] \) given in (8).

**Remark 4.** By the definitions of \( f \), \( m \) and \( M \) we have

\[
0 \leq m[i] \leq f[i] \leq M[i] \leq L - 1.
\]

Further \( M[i] = f[i] \), resp., \( m[i] = f[i] \) if and only if \( w_r[i] = w_g[i] = w_b[i] \), i.e., \( w[i] \) is a gray pixel.

A pixel \( \hat{w}[i] \) has an upper gamut problem if \( \hat{M}[i] > L - 1 \) and
a lower gamut problem if \( \hat{m}[i] < 0 \). We will treat these 
gamut problems in an optimal way in the following sense:

- Assume that we have an upper gamut problem, i.e., \( \hat{M}[i] > L - 1 \) for some \( i \in \mathbb{I}_n \). Then \( \hat{M}[i] = \hat{w}_k[i] \) for some \( k \in \{r, g, b\} \) and the best correction of this pixels is clearly to choose \( a[i] \) in (8) so that \( \hat{w}_k[i] \) has the closest value in the range, i.e., \( \hat{w}_k[i] = L - 1 \), see, e.g., [4]. Equivalently,

\[
L - 1 = a[i] (M[i] - f[i]) + \hat{f}[i].
\]

(11)

From Remark 4 we know that for non gray-valued pixels \( M[i] - f[i] > 0 \), so that

\[
a[i] = \frac{L - 1 - \hat{f}[i]}{M[i] - f[i]} \geq 0.
\]

Thus, for the upper gamut problem, the corrected color values of pixel \( i \) are given by

\[
\hat{w}_c[i] = \frac{L - 1 - \hat{f}[i]}{M[i] - f[i]} (w_c[i] - f[i]) + \hat{f}[i], \quad c \in \{r, g, b\}.
\]

(12)

- Assume we have a lower gamut problem \( \hat{m}[i] < 0 \) for some 
\( i \in \mathbb{I}_n \). Let \( k \in \{r, g, b\} \) be such that \( \hat{w}_k[i] = \hat{m}[i] \). Then the 
optimal correction in (8) obeying (c) is to set \( \hat{w}_k[i] = 0 \), i.e.,

\[
0 = a[i] (m[i] - f[i]) + \hat{f}[i].
\]

(13)

By Remark 4 \( f[i] - m[i] > 0 \) for non gray pixels, so that

\[
a[i] = \frac{\hat{f}[i]}{f[i] - m[i]} \geq 0.
\]

Hence for the lower gamut problem, the corrected color 
value at \( i \) is given by

\[
\hat{w}_c[i] = \frac{\hat{f}[i]}{f[i] - m[i]} (w_c[i] - f[i]) + \hat{f}[i], \quad c \in \{r, g, b\}.
\]

(14)

### A. AFFINE ALGORITHM WITH OPTIMAL RANGE PRESERVATION

Our affine model is a convex combination of the shifting 
and scaling models for some \( \lambda \in [0, 1] \):

\[
\hat{w}_c[i] = \lambda \frac{\hat{f}[i]}{f[i]} w_c[i] + (1 - \lambda) (w_c[i] - f[i] + \hat{f}[i])
\]

(15)

\[
= a[i] (w_c[i] - f[i]) + \hat{f}[i],
\]
where
\[ a[i] := \lambda \frac{\hat{f}[i]}{f[i]} + 1 - \lambda \] (16)

with upper and lower gamut corrections (12) and (14) if necessary. Clearly, for \( \lambda = 1 \) we have the scaling model and for \( \lambda = 0 \) the shifting one. Algorithm 2 corresponds to \( \lambda = 1 \) in (15) but the gamut problem is tackled just by thresholding \( a[i] \) at one; this appears to be an important drawback.

The next propositions show that correcting the gamut problems using (12) or (14) does not yield new gamut problems.

**Proposition 1.** Assume that pixel \( i \in \mathbb{I}_n \) in (15) has an upper gamut problem. Then its correction \( \hat{w}_c[i] \) in (12) satisfies
\[ 0 \leq \hat{w}_c[i] \leq L - 1, \quad c \in \{r, g, b\}. \]

Let us mention that a lower gamut problem can obviously not appear for the multiplicative model (9) i.e. for \( \lambda = 1 \).

**Proposition 2.** Let \( \lambda \in [0, 1) \). Assume that pixel \( i \in \mathbb{I}_n \) in (15) has a lower gamut problem. Then its correction \( \hat{w}_c[i] \) in (14) satisfies
\[ 0 \leq \hat{w}_c[i] \leq L - 1, \quad c \in \{r, g, b\}. \]

Using Propositions 1 and 2, the optimal range-preserving approximation of our affine model (15) can be computed in one iteration where all pixels in the input image are modified only once. The algorithm is described below.

**Algorithm 3 Optimal Range-Preserving Enhancement**
1. Compute the intensity \( f \) of \( w \) by (1) and the target intensity \( \hat{f} \) using Algorithm 1 for given \( \hat{h} \).
2. For \( i \in \mathbb{I}_n \) compute \( M[i] \) and \( m[i] \) by (10). If \( f[i] = 0 \), then \( \hat{w}[i] := 0 \). Otherwise compute
\[ a[i] := \lambda \frac{\hat{f}[i]}{f[i]} + (1 - \lambda), \]
\[ G^\lambda_M[i] := a[i](m[i] - f[i]) + \hat{f}[i], \]
\[ G^\lambda_M[i] := a[i](M[i] - f[i]) + \hat{f}[i], \]
and for all \( c \in \{r, g, b\} \):
(i) \( \hat{w}_c[i] := a[i](w_c[i] - f[i]) + \hat{f}[i] \)
if \( G^\lambda_M[i] \geq 0 \) and \( G^\lambda_M[i] \leq L - 1 \),
(ii) \( \hat{w}_c[i] := \frac{\hat{f}[i]}{M[i] - m[i]}(w_c[i] - f[i]) + \hat{f}[i] \)
if \( G^\lambda_M[i] > L - 1 \),
(iii) \( \hat{w}_c[i] := \frac{\hat{f}[i]}{M[i] - m[i]}(w_c[i] - f[i]) + \hat{f}[i] \)
if \( G^\lambda_M[i] < 0 \).

Algorithm 3 and the role of \( \lambda \) is illustrated in Fig. 3. The images were computed for Gaussian target histogram with parameters \((l, r) = (0.9, 0.1)\), see (27), Sec. V-A.

**B. Multiplicative, Additive Algorithms and their Combinations**

For \( \lambda \in \{0, 1\} \) Algorithm 3 yields two simple range-preserving scaling and shifting algorithms called Multiplicative and Additive algorithm, respectively. Observing that
\[ G^0_M[i] = m[i] - f[i] + \hat{f}[i], \]
\[ G^0_M[i] = M[i] - f[i] + \hat{f}[i], \]
\[ G^1_M[i] = \frac{\hat{f}[i]}{f[i]}M[i] \]
these algorithms read as follows:

**Algorithm 4 Multiplicative Color Enhancement**
1. Compute the intensity \( f \) of \( w \) and the target intensity \( \hat{f} \) using Algorithm 1.
2. For \( i \in \mathbb{I}_n \) compute \( M[i] \) and \( m[i] \) by (10). If \( f[i] = 0 \), then \( \hat{w}[i] := 0 \). Otherwise compute
\[ G^0_M[i] = \hat{f}[i]M[i] \]
and for all \( c \in \{r, g, b\} \):
(i) \( \hat{w}_c[i] := \frac{\hat{f}[i]}{f[i]}w_c[i] \)
if \( G^0_M[i] \leq L - 1 \),
(ii) \( \hat{w}_c[i] := \frac{\hat{f}[i]}{M[i] - f[i]}(w_c[i] - f[i]) + \hat{f}[i] \)
if \( G^0_M[i] > L - 1 \).

**Algorithm 5 Additive Color Enhancement**
1. Compute the intensity \( f \) of \( w \) and the target intensity \( \hat{f} \) using Algorithm 1.
2. For \( i \in \mathbb{I}_n \) compute \( M[i] \) and \( m[i] \) by (10). If \( f[i] = 0 \), then \( \hat{w}[i] := 0 \). Otherwise compute
\[ G^0_M[i] = m[i] - f[i] + \hat{f}[i] \]
and for all \( c \in \{r, g, b\} \):
(i) \( \hat{w}_c[i] := w_c[i] - f[i] + \hat{f}[i] \)
if \( G^0_M[i] \geq 0 \) and \( G^0_M[i] \leq L - 1 \),
(ii) \( \hat{w}_c[i] := \frac{L - 1 - \hat{f}[i]}{M[i] - f[i]}(w_c[i] - f[i]) + \hat{f}[i] \)
if \( G^0_M[i] > L - 1 \),
(iii) \( \hat{w}_c[i] := \frac{L - 1 - \hat{f}[i]}{f[i]}(w_c[i] - f[i]) + \hat{f}[i] \)
if \( G^0_M[i] < 0 \).
Let $\tilde{w}^\times$ be obtained by the Multiplicative algorithm 4 and $\tilde{w}^+$ by the Additive algorithm 4. For some $\lambda \in [0, 1]$, consider
$$\tilde{w}_c := \lambda \tilde{w}^\times + (1 - \lambda) \tilde{w}^+ \quad \forall c \in \{r, g, b\}. \quad (17)$$
Since $\tilde{w}_c$ is a convex combination of $\tilde{w}^\times$ and $\tilde{w}^+$, it obeys all conditions (a)-(c). We want to know if $\tilde{w}_c$ can replace the affine Algorithm 3. In order to answer this question, we set
$$U(\lambda) := \{ i \in \mathbb{N} : G_M^\lambda[i] > L - 1 \},$$
$$L(\lambda) := \{ i \in \mathbb{N} : G_M^\lambda[i] < 0 \}. \quad (18)$$
Here $U(\lambda)$ corresponds to the upper gamut step (ii) and $L(\lambda)$ – to the lower gamut step (iii) in Algorithm 3.

**Proposition 3.** The sets $U(\lambda)$ and $L(\lambda)$ defined in (18) fulfill $U(1) = \emptyset$ and
$$U(\lambda_1) \subseteq U(\lambda_2), \quad L(\lambda_1) \supseteq L(\lambda_2), \quad 0 \leq \lambda_1 < \lambda_2 \leq 1.$$ In particular, (18) yields
$$U(1) = \left\{ i \in \mathbb{N} : \frac{f[i]}{f[i]} M[i] > L - 1 \right\},$$
$$U(0) = \left\{ i \in \mathbb{N} : f[i] - f[i] + M[i] > L - 1 \right\},$$
$$L(0) = \left\{ i \in \mathbb{N} : f[i] - f[i] + m[i] < 0 \right\}. \quad (19)$$
From Proposition 1 one has $U(0) \subseteq U(1)$. The notation in (19) enables Algorithms 4 and 5 to be restated as follows:

- Multiplicative algorithm ($\lambda = 1$)
  (i) $\tilde{w}_c[i] = \frac{f[i]}{f[i]} (w_c[i] - f[i]) + \hat{f}[i]$ if $i \in \mathbb{N} \setminus U(1)$,
  (ii) $\tilde{w}_c[i] = \frac{f[i]}{f[i]} (w_c[i] - f[i]) + \hat{f}[i]$ if $i \in U(1)$.

- Additive algorithm ($\lambda = 0$)
  (i) $\tilde{w}_c[i] = (w_c[i] - f[i]) + \hat{f}[i]$ if $i \in \mathbb{N} \setminus \{U(0) \cup L(0)\}$,
  (ii) $\tilde{w}_c[i] = \frac{f[i] - f[i]}{f[i] - f[i]} (w_c[i] - f[i]) + \hat{f}[i]$ if $i \in U(0)$,
  (iii) $\tilde{w}_c[i] = \frac{f[i] - f[i]}{f[i] - f[i]} (w_c[i] - f[i]) + \hat{f}[i]$ if $i \in L(0)$.

The relation between $\tilde{w}_c$ in (17) and the outcome $\tilde{w}$ of Algorithm 3 for the same $\lambda$ is described in the next proposition.

**Proposition 4.** Let $\tilde{w}$ be obtained by Algorithm 3 and $\tilde{w}$ by (17) for the same $\lambda \in [0, 1]$. Then it holds for $i \in \mathbb{N} \setminus \{U(1) \cup U(0) \cup L(0)\}$ that $\tilde{w}_c[i] = \tilde{w}_c[i]$.

The sets $U(1), U(0)$ and $L(0)$ are usually small for reasonable target intensity images $\hat{f}$ (see Table I) and $U(1) \cup U(0)$ contains generally much less pixels than $U(1)$. If we wish to see the enhancement results $\tilde{w}$ of Algorithm 3 for various $\lambda \in [0, 1]$, Proposition 4 justifies to compute instead $\tilde{w}$ by (17) which is much more practical. Thus, by sliding $\lambda$ in (17), we can easily move between the two models.

**IV. COMPARISON OF THE ALGORITHMS**

**A. Saturation Properties**

Here we analyze the saturation of images enhanced by our methods and by the Naik-Murthy algorithm. The saturation of an RGB image $w$ in the HSI model [5] is defined by
$$S(w) := \begin{cases} 1 - \min\{w_r, w_g, w_b\} f(w) > 0, \\ 0 & \text{if } f(w) = 0. \end{cases} \quad (22)$$

**Proposition 5.** Let $S(w[i])$ and $S(\tilde{w}[i])$ denote the saturation of pixel $i$ in the input image $w$ and the image $\tilde{w}$ obtained by our Algorithm 3, respectively. If $f[i] \in \{m[i], M[i]\}$ we have $S(\tilde{w}[i]) = 0$. Otherwise the obtained saturation is given by
(i) $S(\tilde{w}[i]) = S(w[i]) \left(\lambda + (1 - \lambda) \frac{f[i]}{f[i]} \right)$ if $i \in \mathbb{N} \setminus \{U(\lambda) \cup L(\lambda)\}$,
(ii) $S(\tilde{w}[i]) = S(w[i]) \frac{f[i] - f[i]}{f[i]} L_{-1} - \frac{f[i]}{f[i]}$ if $i \in U(\lambda)$,
(iii) $S(\tilde{w}[i]) = 1$ if $i \in L(\lambda)$.

To clarify the comparison, all magnitudes relevant to Algorithm 2 (Naik and Murthy) hold the superscript $\bullet$, those relevant to Algorithms 4 (Multiplicative) and 5 (Additive) have the superscripts $\times$ and $+$, respectively. In particular, we obtain:

- Algorithm 4 (Multiplicative)
  (i) $\hat{S}(\tilde{w}^\times[i]) = S(w[i])$ if $i \in \mathbb{N} \setminus U(1)$,
  (ii) $S(\tilde{w}^\times[i]) = S(w[i]) \frac{f[i]}{f[i]} \frac{L_{-1} - \hat{f}[i]}{f[i]}$ if $i \in U(1)$.

- Algorithm 5 (Additive)
  (i) $\hat{S}(\tilde{w}^+[i]) = S(w[i]) \frac{f[i]}{f[i]} L_{-1} - \frac{f[i]}{f[i]}$ if $i \in \mathbb{N} \setminus \{U(0) \cup L(0)\}$,
  (ii) $S(\tilde{w}^+[i]) = S(\tilde{w}^\times[i])$ if $i \in U(0)$,
  (iii) $S(\tilde{w}^+[i]) = 1$ if $i \in L(0)$.

Let us denote
$$V := \left\{ i \in \mathbb{N} : \frac{\hat{f}[i]}{f[i]} < 1 \right\}. \quad (25)$$

By (19) and Remark 4 we find that if $i \in U(1)$ then $\frac{\hat{f}[i]}{f[i]} < L_{-1} - m[i] \leq 1$ and that if $i \in L(0)$, then $\frac{\hat{f}[i]}{f[i]} < 1 - m[i] \leq 1$. Hence
$$V \supseteq U(1) \quad \text{and} \quad L(0) \subseteq \mathbb{N} \setminus V. \quad (26)$$

Using the notation in (25), case (i) in Algorithm 2 (Naik-Murthy) holds for any $i \in \mathbb{N} \setminus V$ and step (ii) holds for any $i \in V$. The saturation of images enhanced by applying the Naik-Murthy Algorithm 2 is given by the following proposition.

**Proposition 6.** Let $S(w[i])$ and $S(\hat{w}^\bullet[i])$ denote the saturation of pixel $i$ in the input image $w$ and the image $\hat{w}$ obtained by the Naik-Murthy Algorithm 2, respectively. Then
(i) $S(\hat{w}^\bullet[i]) = S(w[i])$ if $i \in \mathbb{N} \setminus V$,
(ii) $S(\hat{w}^\bullet[i]) = S(w[i]) \frac{f[i]}{f[i]} \frac{L_{-1} - \hat{f}[i]}{f[i]}$ if $i \in V$.

**Remark 5.** Proposition 3 and (26) show that $V \supseteq U(1) \supseteq U(0)$. Note that all these inclusions are always strict; see Table I. E.g., in (26) we find $U(1) = V$ if and only if $M[i] = L - 1$ for all $i \in V$.

Using Propositions 5 and 6, the saturation that Algorithms 4, 5 and 2 provide can be rigorously compared.

- Let $i \in \mathbb{N} \setminus V$. Then $i \notin U(1)$ and $i \notin U(0)$. Hence
  $$S(w[i]) = S(\tilde{w}^\times[i]) = S(\tilde{w}^\bullet[i]) \leq S(\tilde{w}^+[i]),$$
where the last inequality becomes an equality only for \( f[i] = f[i] \). Beyond this case, only the Additive algorithms 5 increases the saturation if \( \hat{f}[i] < f[i] \). But since the output intensity is decreased, the perceived colorfulness of the pixel is decreased.

- Let \( i \in V \setminus U(1) = \{ i \in I_n : 1 < \hat{f}[i] < \frac{M+1}{M} \} \). Then \( i \not\in (U(0) \cup L(0)) \) and consequently

\[
S(w[i]) = S(\hat{w}^x[i]) > S(\hat{w}^+[i]) > S(\hat{w}^*[i])
\]

and \( S(\hat{w}^*[i]) \) decreases faster than \( S(\hat{w}^+[i]) \) when \( \hat{f}[i] \) increases because

\[
S(\hat{w}^*[i]) = \frac{M[i] - f[i]}{L - 1 - f[i]} < 1.
\]

- Let \( i \in U(1) \). Then

\[
S(w[i]) > S(\hat{w}^x[i]) \geq S(\hat{w}^*[i]),
\]

where the equality is reached if and only if \( M[i] = L - 1 \). But for most of the pixels one has \( M[i] < L - 1 \). Further, \( S(w[i]) > S(\hat{w}^+[i]) \geq S(\hat{w}^*[i]) \), where the equality holds if and only if \( M[i] = L - 1 \) and \( i \in U(1) \). So the inequality is strict for most of the pixels.

In all cases, the images enhanced by the Maik-Murthy algorithm have the weakest saturation. On \( I_n \setminus V \), where the target intensity is less than the input intensity, the Additive algorithm 5 gives a better saturation than the Multiplicative algorithm 4. On \( V \setminus U(1) \) the Multiplicative algorithm 4 gives rise to a better saturation than the Additive algorithm 5.

**B. Qualitative comparison**

We begin with a simple but instructive example where we apply Algorithms 4, 5 and 2 to two different “images” each composed of one dark and one bright pixel, resp.,

\[
w_{\text{dark}} = (25, 48, 32), \quad w_{\text{bright}} = (80, 172, 108)
\]

having the same hue but different intensities \( f_{\text{dark}} = 35 \) and \( f_{\text{bright}} = 120 \). In Figs. 4 and 5, the input pixels \( w \) are shown on the top row, while the next rows detail the results of the algorithms w.r.t. the target intensity \( \hat{f} \in \{0, \ldots, 255\} \) given on the \( x \)-axis. By (19) we see that the pixel belongs to \( U(1) \) for \( \hat{f} > \frac{(L-1)f}{M} \) := \( f_{U(1)} \), to \( U(0) \) for \( \hat{f} > (L - 1) - M + f := f_{U(0)} \), to \( L(0) \) for \( \hat{f} < f - m := f_{L(0)} \) and to \( V \) for \( \hat{f} > f := f_{V} \). The corresponding values for our dark and bright image are given in the following table:

<table>
<thead>
<tr>
<th>( w_{\text{dark}} )</th>
<th>( f_{U(1)} )</th>
<th>( f_{U(0)} )</th>
<th>( f_{L(0)} )</th>
<th>( f_{V} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>185.9</td>
<td>242</td>
<td>10</td>
<td>35</td>
<td></td>
</tr>
<tr>
<td>177.9</td>
<td>203</td>
<td>40</td>
<td>120</td>
<td></td>
</tr>
</tbody>
</table>

Fig. 4 deals with the dark pixel \( w_{\text{dark}} \). The Multiplicative algorithm 4 (i) is applied for \( \hat{f} \in [0, 185.9] \). All color values are multiplied by \( \hat{f}/f \), where \( \hat{f}/f > 1 \) for \( \hat{f} > 35 \) which yields a clear increase of the distance between all color channels. The third row shows a pleasant enhancement of the dark input pixel. By (23) the input saturation is preserved. The Additive Algorithm 5 (i) is performed for \( \hat{f} \in [10, 242] \), where all color values are increased by the same amount \( \hat{f} - f \). Since the input pixel is quite dark, the values \( w_c, \ c \in \{r, g, b\} \) and \( f \) are relatively close to each other. For this reason, all color channels \( \hat{w}^c \) remain close to each other. On \([10, 35]\) we have \( f > \hat{f} \) so by (24), \( S(\hat{w}^+) > S(w) \) and \( S(\hat{w}^*) \) continuously decreases from 1 to \( S(w) = 0.29 \). On \([35, 242]\), \( S(\hat{w}^+) \) decreases from \( S(w) = 0.29 \) to 0.04 according to \( S(w)/\hat{f} \). This explains why the colors on the third row remain quite dull, compared to Algorithm 4. For Algorithm 2, case (i) holds only for \( \hat{f} \in [0, 35] \), where the input saturation is preserved. If \( \hat{f} > 35 \), step (ii) is performed and \( S(\hat{w}^*) \) decreases much faster than in Algorithm 5. As a consequence, on \([35, 255]\), the enhanced colors tend to be nearly equal and the obtained color values are nearly gray, see the third row in the figure.

Fig. 5 shows the performance for the brighter pixel \( w_{\text{bright}} \). The Multiplicative algorithm 4 (i) is applied for \( \hat{f} \in [0, 177.9] \). The input saturation is preserved. The Additive algorithm 5 (i) holds for \( \hat{f} \in [40, 203] \). On \([40, 120]\) the recovered saturation decreases from 1 to \( S(w) = 0.33 \) and on \([120, 203]\) it slowly decreases to 0.65\( S(w) \). In Algorithm 2, step (i) holds for \( \hat{f} \leq f = 120 \) where the input saturation is unchanged. Step (ii) is applied for \( \hat{f} \in [120, 255] \) — the interval is not so large as in Fig. 4 and the saturation decreases much less fast to zero. On the 3rd row one sees that the colors obtained with all the three algorithms are quite similar.

**Remark 6.** From Fig. 4, if a dark pixel has a wrong hue (e.g. due to compression or printing artifacts, noise, color cast, etc.), the Multiplicative algorithm 4 can magnify the intensity of this wrong color. If the input image contains a lot of such pixels, the Additive algorithm 5 can be a better choice.

Our conclusions drawn in Subsection IV-A and our findings for one pixel images are confirmed by our tests on the two images, bungalow (underexposed) and flower (slightly lustreless) depicted in Fig. 6 and 7. The distribution of \( \hat{f}[i]/f[i] \) for these two images is very different – the first one ranges on \([0, 18]\) and the second one on \([0, 1.22] \). Roughly speaking, bungalow
mimics the phenomena explained for Fig. 4 and flower and the Naik-Murthy algorithm; it performs better than the last one.

Our algorithms depend, up to a certain degree, on the choice of a target histogram for the intensity channel. Various target histograms avoiding the drawbacks of HE have been proposed in the literature. Some of them leave gaps in the target histogram which can yeild artifacts as in Fig. 2, see e.g. [20], [26], others preserve the input brightness which limits the enhancement of underexposed images, see, e.g. [23], [28].

The models proposed in [25] combine the input image histogram and a uniform histogram using various penalties and parameters. Instead, we adopt a simple and intuitive approach. A common way for histogram based enhancement is to use the histogram of a well exposed example image; see, e.g., [4]. Commercials in photography and image processing software (e.g., Photoshop) mention that well exposed pictures typically have bell-shaped histograms. Based on these advises we focus on target histograms whose shapes are Gaussian functions \( \hat{h}_o \), with domain \([0, L - 1]\), fixed so that

\[
l := \hat{h}_o(0) \leq 1, \quad \max_{x \in [0, L-1]} \hat{h}_o(x) = 1 \quad \text{and} \quad r := \hat{h}_o(L-1) < 1.
\]

A user has to choose two parameters:

- \( l \in (0, 1] \) which is the desired portion of dark pixels;
- \( r \in (0, 1] \) drawing the desired portion of light pixels.

Note that one cannot choose \( l = r = 1 \). Given \( l \in (0, 1] \) and \( r \in (0, 1] \), the shape of the target histogram reads as

\[
\hat{h}_o(x) = \exp \left( - \frac{(x - \mu)}{\sigma} \right), \quad x \in [0, L - 1], \quad \text{for}
\]

\[
\mu = \frac{(L-1)/(\ln L - \ln l)}{\ln r - \ln l}, \quad \sigma = \frac{(L-1)^2(\sqrt{\ln L} - \sqrt{\ln l})^2}{(\ln r - \ln l)^2}.
\]

Finally the target histogram \( \hat{h} \) is normalized according to the number \( n \) of pixels in the image:

\[
\hat{h}(x) = \frac{n \hat{h}_o(x)}{\sum_{x=0}^{L-1} \hat{h}_o(x)} \quad \forall \ x \in \{0, \ldots, L - 1\}.
\]

Whenever (27) is used, we shall write \( \hat{h}_o \) for \( \hat{h} \).

**Remark 7.** If the input RGB image \( w \) has no pixel values on an interval \([0, L_0]\) for some \( L_0 \geq 1 \) one has to perform the hue-preserving stretching in (5). The target histogram is chosen based on the stretched histogram and the enhanced image is computed from the stretched image; see Fig. 16.

With this cautionary remark, we can explain how to choose good target histograms using (27). The input intensity histogram, after stretching if necessary, is denoted by \( h_f \).
Remark 8. The choice of the parameters \((l, r)\) to build \(\hat{h}_o\) in (27) depends on the input intensity histogram \(h_f\) and on the enhancement task. E.g., \((l, r) = (1, 0.99)\) leads to HE.

(i) Function \(\hat{h}_o\) can be easily adapted to all images whose histogram \(h_f\) is roughly unimodal – see the original images in Figs. 6, 7, 9, 10, 11, 13, 14 and 16.

If the pixel values are mainly in the middle of the interval \([0, L - 1]\) and decay at the ends (see Figs. 7, 9 and 16), a good enhancement can be done with \(l \gtrless r \in [0.1, 0.2]\). When \(h_f\) rapidly decays towards \(L - 1\), one should choose \(r \in (0, 0.1]\) (see Figs. 6, 10, 11, 13 and 16). The stronger this decay, the smaller the value of \(r\) should be selected (e.g. in Fig. 11, \(r = 10^{-4}\)).

A too large \(r\) should entail artifacts typical for HE.

If most of the pixel have small values (underexposed images, see Figs. 6, 10, 11 and 13), it is reasonable to select \(l \in [0.8, 1]\). The higher the concentration near 0, the larger the value of \(l \lesssim 1\) should be taken.

(ii) For images with important very dark and very bright areas, function (27) should not work well. Then a good option is to take a mixed target histogram

\[
\hat{h}_{\text{mix}} := \frac{1}{2}(h_f + \hat{h}_o)
\]

(28)

where the parameters \((l, r)\) for \(\hat{h}_o\) are selected following the rules in (i). For example, see Figs. 8, 12 and 15.

This remark is illustrated in Fig. 8.

B. Enhancement Tests

We present some results from a large series of test images with the goal to improve the visual quality. The enhanced image should seem natural and an observer should not suspect that it is a post-processing result.

We compare our Multiplicative algorithm 4 \((\times)\) with the Naik-Murthy algorithm 2 (NM) and the Additive algorithm 5 \((+)\). The HS in these algorithms is done by Algorithm 1.

The histogram of the original intensity image and the target histograms are depicted beneath the images. The percentage of pixels having a gamut problem in Alg. 4 \((\times)\), Alg. 5 \((+)\) and Alg. 2 (NM) is contained in Table I. Further we provide comparison results with

- the fast implementation of ACE by [14] available online at http://demo.ipol.im/demo/g_ace/, and
- the perceptual color enhancement through variational methods in [7] and [8].

ACE has one main parameter, the enhancement strength \(\alpha\) whose default value \(\alpha = 0.5\) is often a good choice. For the two perceptual enhancement methods [7] and [8], the authors gave us their codes and helped us to tune the parameters.

We present the results for images with different defects. The parameter values for all methods are given in the captions, as well as the image credits. The original images in Figs. 6, 7, 8, 10, 11, 13, 12 and 16 are photos taken by the authors who wanted to improve them. For all these images, we do not have “ground truth”. For Fig. 17 we shot an underexposed and a better “example” image which enabled us to compare with the perceptual histogram-based method in [10].
and also some quite clear areas; see we use a mixed target histogram \( \hat{h}_{\ell} \) shift the colors towards blue; observe the stones.

Remark 6). The result in (b) is convincing and the details in of JPEG artifacts so we prefer our Additive algorithm (see enhancement, the dark are clarified. For the ACE in (d) we use a small Remark 8(ii), we take a mixed target histogram. All results give a different atmosphere.

The cathedral photo in Fig. 11(a) is much too dark. The result of our Multiplicative algorithm in (b) is quite colorful. The same algorithm for the mixed target histogram (28) gives in (c) a darker and still colorful image. The results of [7] in (d) and [8] in (c) have a darker color palette. The NM algorithm produces a nearly gray value image shown in (f). All results give a different atmosphere.

The photo taken in Jericoacoara, Fig. 12(a), has very dark and also some quite clear areas; see \( h_f \) in (c). By Remark 8(ii), we use a mixed target histogram \( \hat{h}_{\text{mix}} \). The original has lots of JPEG artifacts so we prefer our Additive algorithm (see Remark 6). The result in (b) is convincing and the details in the dark are clarified. For the ACE in (d) we use a small enhancement, \( \alpha = 3 \), in order to limit the false color shift.

The orchid image in Fig. 13(a) has a bad flashlight effect. This artifact is removed by all tested methods and the background of the scene is clear. Our Multiplicative algorithm gives a realistic colorful result, see (b). As in Fig. 6, our Additive algorithm produces a rather pale image (c) while the issue of the NM algorithm in (d) is too gray. The images obtained by [7] in (e) and by the ACE in (f) exhibit color shifts (see the green leaves on the right and the grass on the bottom left).

The frog image in Fig. 14(a) has an intensity histogram between (i) and (ii) in Remark 8. Indeed, both recipes gave similarly good results. The result with \( \hat{h}_{\ell} \) is shown in (b). For the ACE in (c) we select a small \( \alpha = 3 \) limit the color shift.

The ferrari image in Fig. 15(a) has very dark and very bright areas. Using Remark 8 (ii), we take a mixed target histogram. Our Multiplicative algorithm gives a realistic image shown in (b) with vivid colors that fit the typical red of the brand. The variational methods [7] in (c) and [8] in (d) outperform the ACE (result not shown).

The image fields in Fig. 16(a) was taken through an aircraft porthole. It has no pixels with values in \([0,109]\). By Remark
3, we use in (b) the global hue-preserving stretching. Its further enhancements using our Multiplicative and Additive algorithms give visually the same results, so only the first one is shown in (c). For the ACE in (d) we take $\alpha = 8$ in order to obtain an enhancement strength similar to (c). The JPEG blue artifacts are stronger in (d) compared to (c).

Fig. 13. (a) Image orchid ($768 \times 1024$) with a bad flashlight effect. Enhancement results: (b) Multiplicative algorithm with $\hat{h}_0$ for $(l, r) = (1, 0.1)$; (c) Additive algorithm for the same target histogram; (d) NM algorithm [18] for the same target histogram; (e) Perceptual variational method [7] for $\gamma = 0.2$; (f) ACE [14], default parameter.

Fig. 14. (a) Image frog, $332 \times 300$ (credits: John D. Willson, USGS Amphibian Research and Monitoring Initiative). Enhancement results: (b) Multiplicative algorithm with $\hat{h}_0$ for $(l, r) = (0.4, 0.1)$; (c) ACE for $\alpha = 3$.

Fig. 15. (a) Image ferrari (courtesy of P. Greenspun) of size $235 \times 240$ and histogram of its intensity channel. Enhancement results: (b) Multiplicative algorithm with mixed target histogram for $(l, r) = (1, 0.1)$; (c) Perceptual variational method [7] with default parameters (courtesy of the authors); (d) Perceptual variational method [8] with Michelson’s contrast function and default parameters (courtesy of the authors).

Fig. 16. (a) Image fields ($512 \times 512$) and histogram of its intensity. Enhancement results: (b) Global affine stretching (Remark 3) and intensity histogram; (c) Multiplicative algorithm applied to (b) with Gaussian target histogram for $(l, r) = (0.1, 0.1)$; (d) ACE for $\alpha = 8$ applied to (a).
TABLE I
Percentage of pixels requiring an upper or lower gamut correction. For our Algorithms 4 (×) and 5 (+) the numbers $\%U(1)$, resp., $\%U(0)$, $\%L(0)$ are very small in all tests; this is not the case for $\%V$ in the NM algorithm.

<table>
<thead>
<tr>
<th>Image</th>
<th>$%U(1)$</th>
<th>$%U(0)$</th>
<th>$%L(0)$</th>
<th>$%V$</th>
</tr>
</thead>
<tbody>
<tr>
<td>bungalow $h_0$</td>
<td>1.90</td>
<td>0.94</td>
<td>0</td>
<td>98.15</td>
</tr>
<tr>
<td>island $h_0$</td>
<td>0.89</td>
<td>0.85</td>
<td>0.94</td>
<td>2.93</td>
</tr>
<tr>
<td>club $h_0$</td>
<td>0.98</td>
<td>0.23</td>
<td>0.67</td>
<td>94.61</td>
</tr>
<tr>
<td>club $h_{mix}$</td>
<td>0.33</td>
<td>0.15</td>
<td>1.39</td>
<td>92.61</td>
</tr>
<tr>
<td>boy-on-stones $h_0$</td>
<td>4.76</td>
<td>2.40</td>
<td>0.54</td>
<td>94.15</td>
</tr>
<tr>
<td>cathedral $h_0$</td>
<td>1.04</td>
<td>0.00</td>
<td>1.38</td>
<td>95.71</td>
</tr>
<tr>
<td>Jericoacoara $h_{mix}$</td>
<td>5.98</td>
<td>5.71</td>
<td>0.40</td>
<td>68.26</td>
</tr>
<tr>
<td>orchid $h_0$</td>
<td>0.21</td>
<td>0.10</td>
<td>1.85</td>
<td>87.93</td>
</tr>
<tr>
<td>frog $h_0$</td>
<td>0.09</td>
<td>0.08</td>
<td>6.79</td>
<td>80.04</td>
</tr>
<tr>
<td>ferrari $h_{mix}$</td>
<td>4.40</td>
<td>4.22</td>
<td>0.30</td>
<td>60.7</td>
</tr>
<tr>
<td>fields $h_0$</td>
<td>4.61</td>
<td>2.99</td>
<td>2.06</td>
<td>89.16</td>
</tr>
</tbody>
</table>

Fig. 17. (a) Input underexposed image flag (1500 × 1215) and (b) example better-exposed flag together with the histograms of their intensities. Enhancement results: (c) the color transfer method [10] and (d) our Multiplicative algorithm with the example target intensity histogram in (b).

image. We have applied our Multiplicative algorithm to image (a) using the intensity histogram of (b). The result, shown in (d), is less dull than the example (b). The third row in Fig. 17 depicts a zoom into the area with the raven (right middle). One observes that the bird is fused with part of the background in (c), whereas it is distinguishable in (b) and (d).

VI. CONCLUSIONS AND FUTURE WORK

This work provides the first comprehensive and rigorous presentation of the wide family of histogram specification based affine color assignment models. We have proposed a fast hue and range preserving algorithm. We analyzed the performances of this algorithm and two of its important instances as well as the gamut preserving method in [18].

Many open questions have been raised that we want to answer in our future research. Since our algorithms are fast, extensions to video should be envisaged. We are aware of the broad literature on color enhancement taking both global and local neighborhood of pixels into account see, e.g., [6], [7], [8], [13], [36]. It will be interesting to incorporate such information into our framework. Moreover, we want to take into account other important properties of the human visual system.

VII. APPENDIX

Proof of Proposition 1: A pixel $i \in \mathbb{N}$ in (15) has an upper gamut problem if

$$\frac{\lambda \hat{f}[i] + (1 - \lambda) f[i]}{f[i]} m[i] + (1 - \lambda)(\hat{f}[i] - f[i]) > L - 1.$$  (29)

The upper bound follows from the choice of $\hat{w}_c[i]$, $c \in \{r, g, b\}$ in (12). We focus the lower bound. First we see that (29) implies $\hat{f}[i] \geq f[i]$, since in case $\hat{f}[i] < f[i]$ we would get by replacing $\hat{f}[i]$ by $f[i]$ in the denominator of the quotient and the second summand in (29) the contradiction $\hat{M}[i] > L - 1$. Since $M[i] - f[i] > 0$ the lower bound holds true if and only if

$$L - 1 - \hat{f}[i]w_c[i] - f[i] + \hat{f}[i]M[i] - f[i] \geq 0,$$

$$L - 1 - \hat{f}[i]w_c[i] - (L - 1)f[i] + f[i]M[i] \geq 0$$

which is clearly fulfilled if

$$\hat{f}[i]M[i] - (L - 1)f[i] \geq 0.$$  (30)

By (29) we have

$$\hat{f}[i]M[i] - (L - 1)f[i] > f[i]M[i] - \left(\lambda \hat{f}[i] + (1 - \lambda) f[i]\right)\left(M[i] - f[i]\right) - \hat{f}[i]f[i]$$

$$= \left(\hat{f}[i] - (\lambda \hat{f}[i] + (1 - \lambda) f[i])\right)(M[i] - f[i]).$$

By $\hat{f}[i] \geq f[i]$ the first factor on the right-hand side is $\geq 0$ the second one is $> 0$. Thus, (30) is satisfied. \□

Proof of Proposition 2: A pixel $i \in \mathbb{N}$ in (15) has a lower gamut problem if

$$\frac{\lambda \hat{f}[i] + (1 - \lambda) f[i]}{f[i]} m[i] + (1 - \lambda)(\hat{f}[i] - f[i]) < 0.$$  (31)

The lower bound is clear from the construction of $\hat{w}_c[i]$ in (14). We show the upper bound. Developing (14) yields

$$\hat{w}_c[i] = \frac{\hat{f}[i]w_c[i] - \hat{f}[i]f[i] + \hat{f}[i]f[i] - \hat{f}[i]m[i]}{f[i] - m[i]}$$

$$= \frac{\hat{f}[i]}{f[i] - m[i]} (w_c[i] - m[i]), \quad c \in \{r, g, b\}.$$  (32)

Using (31), one has

$$\hat{f}[i] \frac{\lambda m[i] + (1 - \lambda)f[i]}{f[i]} + (1 - \lambda)(m[i] - f[i]) < 0$$

and hence

$$\hat{f}[i] < \frac{(1 - \lambda)f[i]}{\lambda m[i] + (1 - \lambda)f[i]} (f[i] - m[i]) < f[i] - m[i].$$

Thus, since $0 < w_c[i] - m[i] < L - 1$, we obtain finally

$$\frac{\hat{f}[i]}{f[i] - m[i]} (w_c[i] - m[i]) < L - 1.$$ \□

Proof of Proposition 3: Let $0 \leq \lambda_1 < \lambda_2 \leq 1$. In the upper gamut case we always have $\hat{f}[i] > f[i]$. Then $\lambda_1 \hat{f}[i] + (1 - \lambda_1) f[i] < \lambda_2 \hat{f}[i] + (1 - \lambda_2)f[i]$, hence $G_{\lambda_2}M[i] \leq G_{\lambda_1}M[i]$ and $U(\lambda_1) \subseteq U(\lambda_2)$. Similarly, since $\hat{f}[i] < f[i]$ in the lower gamut case, $\lambda_1 f[i] + (1 - \lambda_1)f[i] > \lambda_2 \hat{f}[i] + (1 - \lambda_2)f[i]$ so
that $G^a_i[i] \geq G^b_i[i]$ and hence $\mathcal{L}(\lambda_1) \geq \mathcal{L}(\lambda_2)$. □

Proof of Proposition 4: Let $i \in \mathbb{N} \setminus \{U(1) \cup U(0)\}$. Then $\hat{w}^+_{\mathcal{C}}[i]$ is given by (20)(i) and since $U(0) \subseteq U(1)$, the value $\hat{w}^+_{\mathcal{C}}[i]$ is given by (21)(i). Consequently,

$$\hat{w}_c[i] = \lambda \frac{f[i]}{f[i]} (w_c[i] - f[i]) + (1 - \lambda)(w_c[i] - f[i]) + \hat{f}_i = \hat{w}_c[i],$$

where the last equality follows by $U(\lambda) \subseteq U(1)$ and Algorithm 3(i).

Let $i \in U(0)$. Then $\hat{w}^+_{\mathcal{C}}[i]$ is given by (21)(ii) and since $U(0) \subseteq U(1)$, $\hat{w}^+_{\mathcal{C}}[i]$ is given by (20)(ii). We have $\hat{w}^+_{\mathcal{C}}[i] = \hat{w}_c[i]$ which shows by $U(0) \subseteq U(\lambda)$ that $\hat{w}_c[i] = \hat{w}_c[i]$. □

Proof of Proposition 5: By (22) we have $S(w[i]) = 1 - \frac{m[i]}{f[i]}$, or equivalently, $f[i] - m[i] = f[i] S(w[i])$. Algorithm 3 computes

$$\hat{w}_c[i] = d[i](w_c[i] - f[i]) + \hat{f}_i,$$

for

$$d[i] = \begin{cases} \frac{\frac{m[i]}{f[i]} + (1 - \lambda)}{\frac{m[i]}{f[i]} + \lambda} & \text{if } i \in U(\lambda) \setminus \{U(0) \cup U(1)\}, \\ \frac{\frac{m[i]}{f[i]} - 1}{\frac{m[i]}{f[i]} + \lambda} & \text{if } i \in U(0), \\ \frac{\frac{m[i]}{f[i]} - 1}{\frac{m[i]}{f[i]} - \lambda} & \text{if } i \in U(1). \end{cases}$$

Since $d[i] > 0$ in all cases, we have

$$S(\hat{w}[i]) = 1 - \frac{1}{f[i]} \left( d[i] \left( \min\{w_v[i], w_g[i], w_b[i]\} - f[i]\right) + \hat{f}_i \right) = 1 - \frac{f[i]}{f[i]} \frac{d[i]}{f[i]} \min\{w_v[i], w_g[i], w_b[i]\} + \frac{f[i]}{f[i]} d[i] - 1 = S(w[i]) f[i]/f[i].$$

Inserting the above values $d[i]$ into (34) finishes the proof. □

Proof of Proposition 6: The case $\hat{f}_i/[f[i]] \leq 1$ follows just as a special case of Proposition 5(i) for $\lambda = 1$. Let $\hat{f}_i > f[i]$. Then case (ii) in Algorithm 2 can be rewritten as (33) for $d[i] := \frac{\frac{m[i]}{f[i]} - 1}{\frac{m[i]}{f[i]} - \lambda}$. Combining this with (34) we are done. □

Acknowledgement: The authors thank M. Bertalmio, E. Provenzi and N. Papadakis for providing their codes corresponding to [7], [8] and [10] and for the discussion. We want to thank the anonymous reviewers for their constructive remarks.

REFERENCES


Gabriele Steidl received her PhD and Habilitation in Mathematics from the University of Rostock, Germany, in 1988 and 1991, respectively. From 1993 to 1996, she held a position as assistant professor of Applied Mathematics at the University of Darmstadt, Germany. From 1996 to 2010, she was full professor at the Department of Mathematics and Computer Science, University of Mannheim, Germany. Since 2011, she is Professor at the Department of Mathematics at the University of Kaiserslautern, Germany. She was a visiting researcher at the University of Debrecen (Hungary), the University of Zrich (Switzerland), the Banach Center Warsaw (Poland), the ENS Cachan and the University Paris Est (France). Her research interests include applied and computational harmonic analysis and convex analysis with applications in image processing.

Mila Nikolova received the Ph.D. degree from the Université de Paris-Sud, France, in 1995. She got a Habilitation to direct research in 2006. Currently, she is Research Director with the National Center for Scientific Research (CNRS), France. She performs her duty as a full-time researcher at CMLA (Centre de Mathématiques et de Leurs Applications), Department of Mathematics, ENS Cachan, France. Her research interests are in mathematical Image and signal reconstruction, Inverse problems, Regularization and variational methods and the properties of their solutions, Scientific computing.