A fast algorithm for exact histogram speci cation. Simple extension to colour images

Mila Nikolova

A fast algorithm for exact histogram specification. Simple extension to colour images

Mila Nikolova

CMLA CNRS ENS Cachan
61 av. du Président Wilson, 94235 Cachan Cedex, France
nikolova@cmla.ens-cachan.fr
http://mnikolova.perso.math.cnrs.fr/

Abstract. *In [12] a variational method using $\mathcal{C}^2$-smoothed ℓ_1-TV functionals was proposed to process digital (quantized) images so that the obtained minimizer is quite close to the input image but its pixels are all different from each other. These minimizers were shown to enable exact histogram specification outperforming the state-of-the-art methods [6], [19] in terms of* faithful total strict ordering. *They need to be computed with a high numerical precision. However the relevant functionals are difficult to minimize using standard tools because their gradient is nearly flat over vast regions.*
Here we present a specially designed fixed-point algorithm enabling to attain the minimizer with remarkable speed and precision. This variational method applied with the new proposed algorithm is actually the best way (in terms of quality and speed) to order the pixels in digital images. This assertion is corroborated by exhaustive numerical tests.
We extend the method to color images where the luminance channel is exactly fitted to a prescribed histogram. We propose a new fast algorithm to compute the modified color values which preserves the hue and do not yield gamut problem. Numerical tests confirm the performance of the latter algorithm.

Key words: Color image enhancement; Exact histogram specification; Fast smooth convex nonlinear minimization; Fixed point algorithm; Gamut preservation; Hue preservation; Minimizer analysis; Smoothed ℓ_1-TV functionals; Total strict ordering; Variational methods.

1 Introduction

Histogram processing is a technique with numerous applications. The goal of exact histogram specification (HS) is to transform an input image into an output image having a prescribed histogram. Histogram equalization (HE) is a particular case of HS. Among the applications of HS let us mention invisible watermarking, image normalization and enhancement, object recognition [7], [5], [16]. Let f be an input $M \times N$ digital image with L gray values. The set of values of f is denoted by[1] $\mathcal{Q} = \{q_1, \cdots, q_L\}$. To simplify the notation we reorder the

---

[1] For 8-bit images we have $L = 256$ and $\mathcal{Q} = \{0, \cdots, 255\}$.

image columnwise into a vector of size $n := MN$ and address the pixels by the index set[2] $\mathbb{I}_n := \{1, \cdots, n\}$. The histogram of f, denoted by h_f, is given by $h_f[q_k] = \sharp\{i \in \mathbb{I}_n \mid f[i] = q_k\}$, $\forall\, k \in \mathbb{I}_L$, where $\sharp$ stands for cardinality.

Exact HS is straightforward for images whose pixels values are all different from each other. However exact HS (and also exact HE) is an ill-posed problem for digital (quantized) images since the number of pixels[3] n is much larger than number the possible intensity levels L [6] ,[17]. The clue to achieving *exact* HS is to obtain a *meaningful* total strict ordering of all pixels in the input digital image. Research on this problem has been conducted for four decades already [8]. The Local Mean (LM) method of Coltuc, Bolon and Chassery [6], the wavelet-based approach (WA) of Wan and Shi in [19] and the specialized variational approach (SVA) of Nikolova, Wen and Chan [12] are the state-of-the-art methods. For any input pixel $f[i]$ in the input digital image f these methods extract K auxiliary information, say $a_k[i]$, $k \in \mathbb{I}_K$, based on f. For simplicity, we set $a_0 := f$. Then an ascending order "$\prec$" for all pixels is sought using the rule

$$i \prec j \quad \text{if} \quad f[i] \leq f[j] \quad \text{and} \quad a_k[i] < a_k[j] \quad \text{for some} \quad k \in \{0, \cdots, K\}. \quad (1)$$

The numerical results in [12] have shown that SVA clearly outperforms its main competitors—LM and WA—in terms of quality and memory requirements but not in speed. In section 3 we derive a specialized fixed point minimization algorithm that attains the minimizer with remarkable speed and precision. Convergence and parameter selection are briefly discussed. Numerical tests confirm that the SVA method along with the new FP algorithm outperforms by far all other relevant sorting methods.

In section 4 we focus on HS for color digital images. Extension of gray scale HS to color images is a quite complex task. As usual, a color image has three components: red (R), green (G) and blue (B). Applying HS to each color channel independently changes the hue of the image [17]. To avoid this problem, several ways to define a 3-D color histogram were proposed, e.g. [18], [10]. Recently, Han et al. [9] showed that these methods increase the brightness of the image and cannot fit the prescribed (uniform) histogram. In the same article, the authors propose to equalize the luminance (intensity) component of the image and apply the hue-preserving transformation proposed by Naik and Murthy [11] to assign the new color values. There are many methods that rely on modification of the histogram of the luminance component and deduce the needed change in the RGB space, see e.g. [2], [1], [16]. Our approach is to produce a correct template for the luminance part by HS. To compute the color components, we propose a new algorithm preserving the hue and the gamut, and ensuring that the resultant luminance component fits the specified histogram. The new algorithm share the same simplicity as the one used in [9] but provides much better results.

---

[2] In what follows, $\mathbb{I}_m := \{1, \cdots, m\}$ for any integer m.
[3] E.g. for an 1024×1024 8-bit image we have $n = 1048576 \gg 256 = L$.

2   The specialized variational approach (SVA)

The functionals proposed in [12] are of the form

$$J(u,f) := \Psi(u,f) + \beta\Phi(u), \quad \beta > 0 \tag{2}$$

with

$$\begin{aligned}
\Psi(u,f) &:= \sum_{i \in \mathbb{I}_n} \psi(u[i] - f[i]), \\
\Phi(u) &:= \sum_{i \in \mathbb{I}_r} \varphi(g_i u)
\end{aligned} \tag{3}$$

where $g_i \in \mathbb{R}^{1 \times n}$, $i \in \mathbb{I}_r$ correspond to a forward discretization. More precisely,

- If only vertical and horizontal differences are considered

$$\begin{array}{llll}
g_i[i] = -1, & g_i[i+1] = 1 & \text{and} & g_i[k] = 0 \quad \forall\, k \in \mathbb{I}_n \setminus \{i, i+1\}, \\
g_j[j] = -1, & g_j[j+M] = 1 & \text{and} & g_i[k] = 0 \quad \forall\, k \in \mathbb{I}_n \setminus \{j, j+M\};
\end{array} \tag{4}$$

- If diagonal differences are added, $\Phi(u)$ is nearly rotationally invariant and

$$\begin{array}{lll}
g_i[i] = -1, & g_i[i+M-1] = 1 & \text{and} \quad g_i[k] = 0 \quad \forall\, k \in \mathbb{I}_n \setminus \{i, i+M-1\}, \\
g_j[j] = -1, & g_j[j+M+1] = 1 & \text{and} \quad g_j[k] = 0 \quad \forall\, k \in \mathbb{I}_n \setminus \{j, j+M+1\}.
\end{array}$$

In both cases, Neumann or periodic boundary conditions are adopted. We denote

$$G = \left[g_1^T, \cdots, g_r^T \right]^T \in \mathbb{R}^{r \times n},$$

where the superscript T stands for transposed.

The functions $\psi(\cdot) := \psi(\cdot, \alpha_1) : \mathbb{R} \to \mathbb{R}$ and $\varphi(\cdot) := \varphi(\cdot, \alpha_2) : \mathbb{R} \to \mathbb{R}$ depend on two parameters $\alpha_1 > 0$ and $\alpha_2 > 0$, respectively. When necessary, we shall use the notation $\psi(\cdot, \alpha_1)$ and $\varphi(\cdot, \alpha_2)$. The functions ψ and φ in (3) belong to the *family of functions* $\theta(\cdot, \alpha) : \mathbb{R} \to \mathbb{R}$, $\alpha > 0$, satisfying the conditions H1 and H2 described below. We denote $\theta'(t, \alpha) := \frac{d}{dt}\theta(t, \alpha)$, and similarly for θ''.

H 1 *For any $\alpha > 0$ fixed, $\theta(\cdot, \alpha)$ is $\mathcal{C}^s$-continuous for $s \geq 2$, even—i.e. $\theta(-t, \alpha) = \theta(t, \alpha)$—and meets*

$$t \in \mathbb{R} \quad \Rightarrow \quad \theta''(t, \alpha) > 0 .$$

Note that by H1, for any α fixed, $t \to \theta'(t, \alpha)$ *is strictly increasing in* t. Further,

H 2 *For any $\alpha > 0$ given, $\theta'(t, \alpha)$ is upper bounded[4] and for $t > 0$ fixed, it is strictly decreasing in $\alpha > 0$ with*

$$\begin{aligned}
\alpha > 0 \quad &\Rightarrow \quad \lim_{t \to \infty} \theta'(t, \alpha) = 1 , \\
t \in \mathbb{R} \quad &\Rightarrow \quad \lim_{\alpha \to 0} \theta'(t, \alpha) = 1 \quad \text{and} \quad \lim_{\alpha \to \infty} \theta'(t, \alpha) = 0 .
\end{aligned}$$

Under these assumptions, the functional $J(\cdot, f)$ in (2)-(3) is clearly a fully smoothed ℓ_1-TV model. Good choices for θ meeting H1 and H2 are given in Table 1.

Remark that θ'' is even, positive and its upper bound is finite and

$$\|\theta''\|_\infty = \theta''(0) > 0.$$

---

[4] The upper bound of θ' is set to 1 only for definiteness.

	θ	θ'	θ''
f1	$\sqrt{t^2 + \alpha}$	$\dfrac{t}{\sqrt{t^2 + \alpha}}$	$\dfrac{\alpha}{\left(\sqrt{t^2 + \alpha}\right)^3}$
f2	$\alpha \log\left(\cosh\left(\dfrac{t}{\alpha}\right)\right)$	$\tanh\left(\dfrac{t}{\alpha}\right)$	$\dfrac{1}{\alpha}\left(1 - \left(\tanh\left(\dfrac{t}{\alpha}\right)\right)^2\right)$
f3	$\|t\| - \alpha \log\left(1 + \dfrac{\|t\|}{\alpha}\right)$	$\dfrac{t}{\alpha + \|t\|}$	$\dfrac{\alpha}{(\alpha + \|t\|)^2}$

Table 1: Relevant choices for $\theta(\cdot, \alpha)$ obeying H1 and H2. When $\alpha > 0$ decreases towards zero, $\theta(\cdot, \alpha)$ becomes stiff near the origin.

2.1   Preliminary facts

Using H1 and H2, the properties listed below play a role in what follows.

1. For any $\beta > 0$ and any f, $J(\cdot, f)$ has a unique minimizer $\hat{u}$ [12, Proposition 1].
2. For any $\beta > 0$ and any f living in a *dense open subset of* $\mathbb{R}^n$, say $\mathbb{K}^n$, the minimizer $\hat{u}$ of $J(\cdot, f)$ satisfies [12, Theorem 1]

$$\begin{aligned} \hat{u}[i] \neq \hat{u}[j], &\quad \forall\, i, j \in \mathbb{I}_n, \ \ i \neq j; \\ \hat{u}[i] \neq f[i], &\quad \forall\, i \in \mathbb{I}_n. \end{aligned} \tag{5}$$

   However, all digital images with L gray values (like f) belong to a subset $S_{\mathcal{Q}}^n$ which is closed and of null Lebesgue measure in $\mathbb{R}^n$. Using some results from number theory, the conclusion drawn in [12, sect. 2, Remark (b)] is that $\sharp\left(\mathbb{K}^n \cap S_{\mathcal{Q}}^n\right)/\sharp\, S_{\mathcal{Q}}^n$ should be a number close to zero[5]. Then the minimizer $\hat{u}$ of $J(\cdot, f)$ for $f \in S_{\mathcal{Q}}^n$ satisfies (5) with a very high probability. Thus $\hat{u}$ provides the auxiliary information to strictly order the pixels in f using (1).
3. Since $\psi'(\cdot, \alpha)$ is $\mathcal{C}^{s-1}$ and odd, it has an inverse function

$$\xi(\cdot, \alpha_1,) := (\psi')^{-1}(\cdot, \alpha) : (-1, 1) \to \mathbb{R}, \tag{6}$$

   which is also odd, strictly increasing and $\mathcal{C}^{s-1}$ (inverse functions theorem).
4. For any $y \in (0, 1)$, the function $\alpha \mapsto \xi(y, \alpha)$ is strictly increasing on $(0, +\infty)$ [3, Lemma 2].
5. Let us denote $\eta := \|G\|_1$. If $\beta\eta < 1$ then $\|\hat{u} - f\|_\infty \leq \xi(\beta\eta, \alpha_1)$. and $\alpha_1 \mapsto \xi(\beta\eta, \alpha_1)$ is strictly increasing on $(0, +\infty)$.
6. Further, $\|\hat{u} - f\|_\infty \nearrow \xi(\beta\eta, \alpha_1)$ as $\alpha_2 \searrow 0$ [3, Theorem 2].

3   A fast sorting algorithm

3.1   Semi-Explicit Formula for the Minimizer

The unique minimizer $\hat{u}$ of $J(\cdot, f)$ satisfies $\nabla J(\hat{u}, f) = 0$ where the gradient ∇ is taken with respect to the first variable, namely u. From the definition of J in

---

[5] Note that no reasonable sorting algorithm can order strictly the pixels of *all* digital images. E.g., the pixels of a constant image should not be ordered in a strict way.

(2) this is equivalent to $\nabla \Psi(\hat{u}, f) = -\beta \nabla \Phi(\hat{u})$. Using (3), we have

$$\frac{d\Psi(u, f)}{du[i]} = \psi'(u[i] - f[i]) \quad \text{and} \quad \frac{d\Phi(u)}{du[i]} = \sum_{j \in \mathbb{I}_r} \varphi'(g_j u) g_j[i]. \tag{7}$$

Thus the minimizer $\hat{u}$ satisfies

$$\psi'(\hat{u}[i] - f[i]) = -\beta \sum_{j \in \mathbb{I}_r} \varphi'(g_j \hat{u}) g_j[i], \quad \forall i \in \mathbb{I}_n.$$

Using the notation in (6), the latter equations are equivalent to

$$\hat{u}[i] = f[i] + \xi \left(-\beta \sum_{j \in \mathbb{I}_r} \varphi'(g_j \hat{u}) g_j[i] \right), \quad i \in \mathbb{I}_n. \tag{8}$$

The inverse function $\xi(y, \alpha) = (\theta')^{-1}(y, \alpha)$ in (6) has an explicit expression for f1, f2 and f3 in Table 1. This function and its derivative $\xi' := \frac{d}{dy}\xi(y, \alpha)$ are given in Table 2. Note that ξ' is even and strictly increasing on $[0, 1)$.

	ξ	ξ'
f1	$y\sqrt{\dfrac{\alpha}{1 - y^2}}$	$\dfrac{\sqrt{\alpha}}{(\sqrt{1 - y^2})^3}$
f2	$\dfrac{\alpha}{2} \ln \dfrac{1 + y}{1 - y}$	$\dfrac{\alpha}{1 - y^2}$
f3	$\dfrac{\alpha y}{1 - |y|}$	$\dfrac{\alpha}{1 - |y|}$

Table 2: The inverse function $\xi(y, \alpha) = (\theta')^{-1}(y, \alpha)$ in (6) and its derivative ξ' with respect to y for all functions in Table 1.

3.2 A fixed point (FP) algorithm to minimize $J(\cdot, f)$

The proposed algorithm uses (8) and Table 2. The iterations are given by

$$u_{k+1} = \mathcal{X}(u_k), \tag{9}$$
$$\mathcal{X}(u) := f + \xi\big(-\beta \nabla \Phi(u)\big), \tag{10}$$

where the function ξ, given in Table 2, is applied componentwise and $u_0 = f$.

Theorem 1. *Let α_1, α_2 and β be chosen so that $\beta\eta < 1$ and*

$$\beta \, \xi'(\beta\eta) \, \varphi''(0, \alpha_2) \, \|G^T G\|_\infty < 1, \tag{11}$$

where η is defined in 5, subsection 2.1. Then the iteration (9)-(10) converges.

Sketch of the proof. For any $\alpha_2 > 0$,

$$0 < \varphi''(g_i u, \alpha_2) \leq \varphi''(0, \alpha_2) \quad \forall \, i \in \mathbb{I}_r.$$

Further, one derives

$$\|\nabla\mathcal{X}(u)\|_\infty \leq \beta\xi'(\beta\eta)\|G^T \mathrm{diag}\big(\varphi''(g_i u)\big)G\|_\infty \leq \beta\xi'(\beta\eta)\varphi''(0)\|G^T G\|_\infty.$$

Then $\|\nabla\mathcal{X}(u)\|_\infty < 1$. The spectral radius of $\nabla\mathcal{X}(u)$ meets $\rho(\nabla\mathcal{X}(u)) \leq \|\nabla\mathcal{X}(u)\|_\infty$. Since $\mathcal{X}$ has a fixed point by (8), Ostrowski theorem [13] entails the result. □

Some practical values of the parameters ensuring convergence are given next.

- If G corresponds only to (4), then $\eta = \|G\|_1 = 4$ and $\|G^T G\|_\infty = 8$. For ψ and φ given by f1 in Table 1 and $\alpha_1 = 0.05$, $\alpha_2 = 0.3$ and $\beta = 0.1$ we have

$$\|\nabla\mathcal{X}(u)\|_\infty \leq 0.4242 \quad \text{and} \quad \|\hat{u} - f\|_\infty \lessapprox 0.0976.$$

- If G corresponds to jointly (4) and (2), $\eta = \|G\|_1 = 8$ and $\|G^T G\|_\infty = 16$. For ψ and φ given by f1 and $\alpha_1 = 0.02$, $\alpha_2 = 0.4$ and $\beta = 0.07$ we have

$$\|\nabla\mathcal{X}(u)\|_\infty \leq 0.6963 \quad \text{and} \quad \|\hat{u} - f\|_\infty \lessapprox 0.0956.$$

Remark 1. When initialized with a nonconstant image, the iteration (9)-(10) provides fast convergence even if (11) is not satisfied. One of the reason is that for many differences we have $\varphi''(g_i u) > 0$ in which case $\|G^T \mathrm{diag}\big(\varphi''(g_i u)\big)G\|_\infty \ll \varphi''(0)\|G^T G\|_\infty$. And when $\|\mathcal{X}(u_0) - \mathcal{X}(u_1)\| < 1$, then the iteration converges (see [15, p. 142]). Another reason is that $\rho(\nabla\mathcal{X}(u))$ is quite smaller than $\|\nabla\mathcal{X}(u)\|_\infty$ and so under the condition in (11), $\rho(\nabla\mathcal{X}(u))$ is quite smaller than 1.

3.3   Comparison with the state-of-the-art sorting algorithms

The variational method provides one auxiliary information which is the minimizer $\hat{u}$ of $J(\cdot, f)$, i. e. $a_1[i] = \hat{u}[i] \; \forall \, i \in \mathbb{I}_n$ and ordering is obtained by (1). As in [12], $J(\cdot, f)$ was used with G corresponding to (4) and

$$\psi(t) = \sqrt{t^2 + \alpha_1}, \quad \varphi(t) = \sqrt{t^2 + \alpha_2}, \quad \alpha_1 = \alpha_2 = 0.05 \quad \text{and} \quad \beta = 0.1.$$

We ran the Polak-Ribière (PR) CG minimization with stopping rule given by $\|\nabla J(u_k, f)\|_\infty \leq 10^{-6}$ and limiting the iteration number to 35, as in [12]. Our FP algorithm was applied with stopping rule $\|\nabla J(u_k, f)\|_\infty \leq 10^{-6}$. Our method was compared with the local mean (LM) algorithm [6] for $K = 6$ and with the wavelet-based algorithm (WA) [19] for Haar wavelet for $K = 9$. These values of K were recommended by the authors. The experiments were performed using a PC DELL Latitude E6220 with an Intel Core i7-2640M, 2.8 GHZ processor and 8 GB of RAM under Windows 7, using MATLAB v. 7.11.0.584, 64-bit.

Here we present sorting results on 12 digital images with various sizes and content, with gray values in $\{0, \cdots, 255\}$. The images and their histograms are shown in Fig. 1. Note that most of these histograms are quite singular.

airplane(512^2)   baril (512^2)   car (512^2)   clock (512^2)   couple (256^2)   F-16 (512^2)

house (512^2)   Lena (512^2)   man (1024^2)   raffia (512^2)   sand (512^2)   stream (512^2)
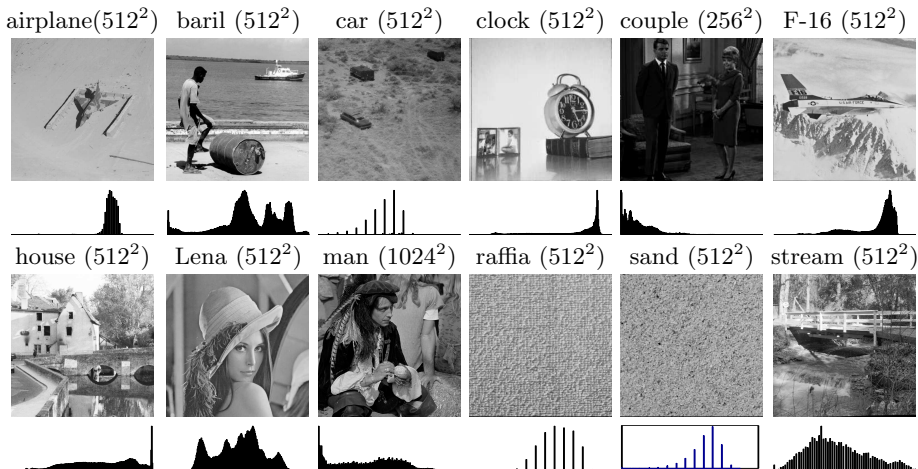

Fig. 1: All 12 digital 8 bit images used to compare the algorithms and their histograms. The gray values of these images belong to $\{0, \cdots, 255\}$.

Remark 2. Since our parameter choice guarantees that $\|\hat{u} - f\|_\infty < 0.1$ (see 5 in subsection 2.1), ordering the pixels according to (1) amounts just to sort $(\hat{u} + f)$.

This fact was not noticed in [12] where (1) was used directly which is computationally heavier. For the LM and WA methods, (1) must be applied for $K = 6$ and $K = 9$ images, respectively, which requires much more memory and computation than the SVA method [12] where $K = 1$.

The pixel ordering provided by the PR minimization in [12] and the new FP algorithm should be the same since the obtained PSNR values for HE inversion are the same (these experiments are not presented here). The experiments in [12, section 5] have shown that SVA outperforms by far the LM and WA methods in terms of PSNR in restoration of contrast compression and in HE inversion. The proposed FP minimization scheme gives rise to a much shorter CPU time and a more than 5 times better numerical precision. For fair comparison of the numerical schemes, Remark 2 was not used to generate the results in Table 3. When Remark 2 is applied, the mean CPU time for the SVA-FP algorithm is reduced to **1.54 sec.** *In terms of faithful total strict ordering and CPU time, the SVA with the proposed FP scheme and using the simple ordering rule in Remark 2 provides the best results.* The same conclusion was drawn on a test on 50 eight-bit gray-value images downloaded from http://sipi.usc.edu/database/.

4   HS for color images

4.1   Our approach

Let $w = (w_1, w_2, w_3)$ be an input color image where w_1, w_2 and w_3 are its red, green and blue channels, respectively. Let $\zeta = (\zeta_1, \cdots, \zeta_L)$ be the prescribed

	Fail %			CPU				SVA	
Image	LM	WA	**SVA**	LM	WA	SVA	SVA **FP**	$\|\nabla J\|_\infty \times 10^{-7}$ PR	**FP**
airplane	5.30	17.70	0.00	2.85	6.29	2.54	2.07	45.70	6.06
baril	0.17	0.24	0.00	2.15	5.34	2.37	1.92	3.83	6.45
car	7.84	19.91	0.00	3.67	6.51	2.28	2.22	5.96	7.30
clock	1.57	4.52	0.00	1.05	2.20	0.69	0.44	4.47	6.92
couple	2.50	3.30	0.00	0.94	2.11	0.53	0.37	6.68	7.38
F-16	0.18	0.57	0.00	2.48	5.21	5.21	2.11	63.19	7.22
houseB	0.36	1.58	0.00	1.62	5.40	2.40	2.37	15.02	5.71
Lena	0.00	0.20	0.00	2.84	4.91	4.57	1.67	58.95	5.81
man	0.34	0.68	0.00	7.58	15.68	9.24	9.38	29.13	7.65
raffia	13.66	16.05	0.00	2.71	6.99	2.37	1.87	10.60	5.43
sand	12.62	15.21	0.00	3.82	6.63	4.82	2.45	68.90	5.80
stream	0.41	0.75	0.00	2.75	4.98	2.34	2.29	6.08	7.22
means	2.97	5.28	0.00	2.19	4.46	2.46	1.79	20.23	4.88

Table 3: Comparison with the state-of-the-art algorithms. Fail denotes the percentage of pixels that could not be sorted in a strict way. CPU is in seconds.

histogram. As in the previous section, we consider that all w_k's are reordered columnwise as n-length vectors. The luminance of w is [4]

$$f = \frac{1}{3}(w_1 + w_2 + w_3) \in \mathbb{R}^n.$$

With the help of the ordering algorithm described in section 3, f is transformed into $\widehat{f} \in \mathbb{R}^n$ whose histogram is exactly ζ. We need a color image $\widehat{w}$ such that

$$\frac{1}{3}(\widehat{w}_1 + \widehat{w}_2 + \widehat{w}_3) = \widehat{f} \tag{12}$$

and satisfying the classical requirements:

(c1) $\widehat{w}$ has the same hue as w;
(c2) to avoid gamut problem, $0 \le \widehat{w}_k[i] \le L - 1$, $\forall i \in \mathbb{I}_n$, $k = 1, 2, 3$.

It is well known (and easy to verify) that the hue of a pixel $w[i]$ is guaranteed to be preserved in the restored $\widehat{w}[i]$ only if $\widehat{w}[i]$ is obtained from $w[i]$ using an affine transform [4], [11].

Our method to compute the color channels from w and $\widehat{f}$ consists in a "slight" but important modification of the method proposed in [11] and used in [2], [9], among others. It is composed of a forward step followed by a correction step.

Algorithm for color assignment

(step 1) Compute $F \in \mathbb{R}^n$ according to

$$F[i] := \frac{\widehat{f}[i]}{f[i]} \quad \forall i \in \mathbb{I}_n$$

and assign

$$\widehat{w}_k[i] = F[i]\, w_k[i] \quad k = 1, 2, 3 \quad \forall\, i \in \mathbb{I}_n. \tag{13}$$

(step 2) Find the set

$$\mathcal{J} := \{i \in \mathbb{I}_n \mid \widehat{w}_1[i] > L - 1 \text{ or } \widehat{w}_2[i] > L - 1 \text{ or } \widehat{w}_3[i] > L - 1\}. \tag{14}$$

Compute $C \in \mathbb{R}^{\sharp \mathcal{J}}$ by

$$C[i] := \frac{L - 1 - \widehat{f}[i]}{L - 1 - f[i]} \quad \forall\, i \in \mathcal{J}$$

and correct the pixels in $\mathcal{J}$ as

$$\widehat{w}_k[i] = L - 1 - C[i]\,(L - 1 - w_k[i]) \quad k = 1, 2, 3 \quad \forall\, i \in \mathcal{J}. \tag{15}$$

It is easy to check that that both modifications in (13) and (15) satisfy (12). A value $F[i] > 1$ means that the color at pixel i should be enhanced, i.e., $|w_k[i] - w_{k'}[i]| < |\widehat{w}_k[i] - \widehat{w}_{k'}[i]|$, $k \neq k'$, $k, k' \in \{1, 2, 3\}$. So we wish to keep the maximum number of pixels computed using (13). Some of them (quite a few in practice) will fail the constraint (c2)—these pixels form the set $\mathcal{J}$ in (14). Their value will be properly modified at the correction step 2.

Remark 3. In the scheme of Naik and Murthy [11] (and the one of [16]) , step 1 is applied only if $F[i] \leq 1$ and in all other cases step 2 is applied. We can note that their strategy is quite conservative. For instance, if $w[i] = (10, 30, 50)$ and $F[i] = 5$, noticing that $f[i] = 30$, their strategy yields $\widehat{w}[i] = (140.67, 150, 159.33)$ which results in a nearly gray-value pixel. Instead, our approach yields $\widehat{w}[i] = (50, 150, 250)$, so the color is enhanced and the constraint (c2) is satisfied.

The computational cost of the algorithm of Naik and Murthy was analysed in [9]. The conclusion was that the computational complexity is proportional to the number of pixels. In all experiments, we observed that in mean 3.5 % of the pixels go through step 2. So the computational cost of our color assignment algorithm is nearly the same.

4.2   Numerical results

Here we compare our algorithm for color assignment with the algorithm proposed in [11] and used in [9]. For fair comparison of the color assignment algorithms, *in all cases we used our sorting algorithm (section 3)*. Exhaustive comparison with other algorithms for HE of color images can be found in [9].

The original image $(800 \times 800 \times 3)$ in Fig. 2 is underexposed and has a poor contrast. HE often produces overly enhanced unnatural looking images. The target histogram was chosen according to general recommendations of commercials in image processing (seen on You Tube). It is exactly satisfied and can be seen on the last row of the histograms of the restored images (in black). The image obtained by [11], [9] suffers from being too gray. This confirms our Remark 3.
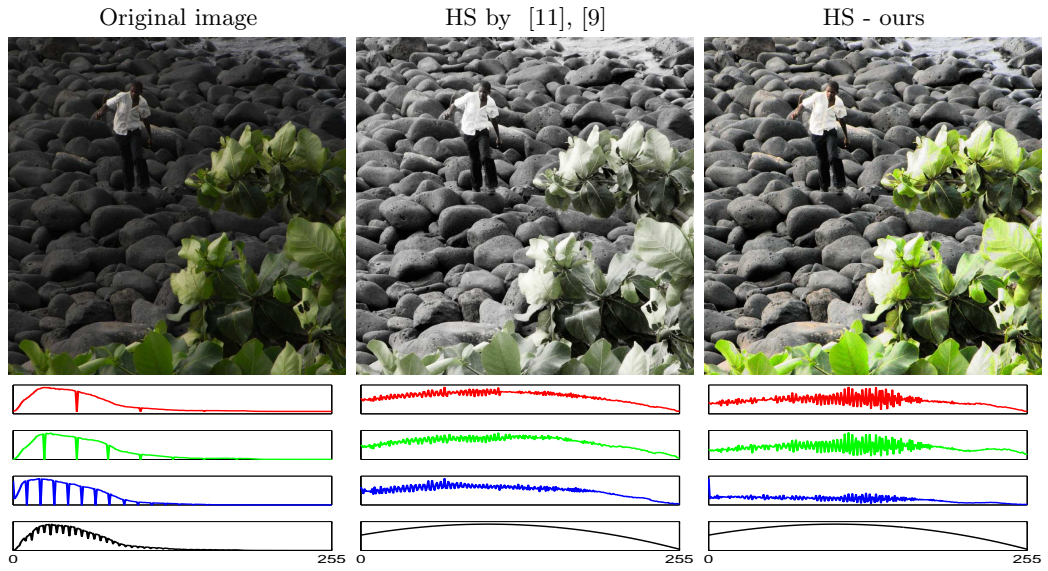
Original image   HS by [11], [9]   HS - ours

Fig. 2: Images and their histograms—R (■), G (■), B (■), luminance (■).

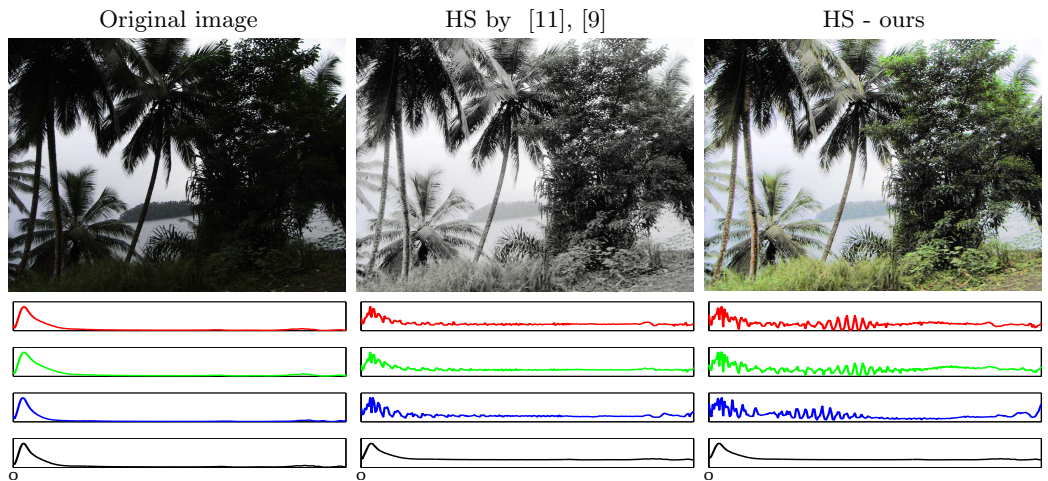Original image   HS by [11], [9]   HS - ours

Fig. 3: Images and their histograms—R (■), G (■), B (■), luminance (■).

Using our algorithm (13)-(15)—only 4.7 % of the pixels needed the correction step 2. The image quality is really improved.

In Fig. 3 the original image ($750 \times 1000 \times 3$) seems nearly gray-valued. Following [1], the prescribed histogram is a linear combination of the histogram of the input image and a uniform histogram—see the last row of the histograms of the restored images. The image obtained by [11], [9] is almost gray-valued. Our method enables us to recover all colors. In this case, only 1.27 % of the pixels had to be rescaled using step 2.
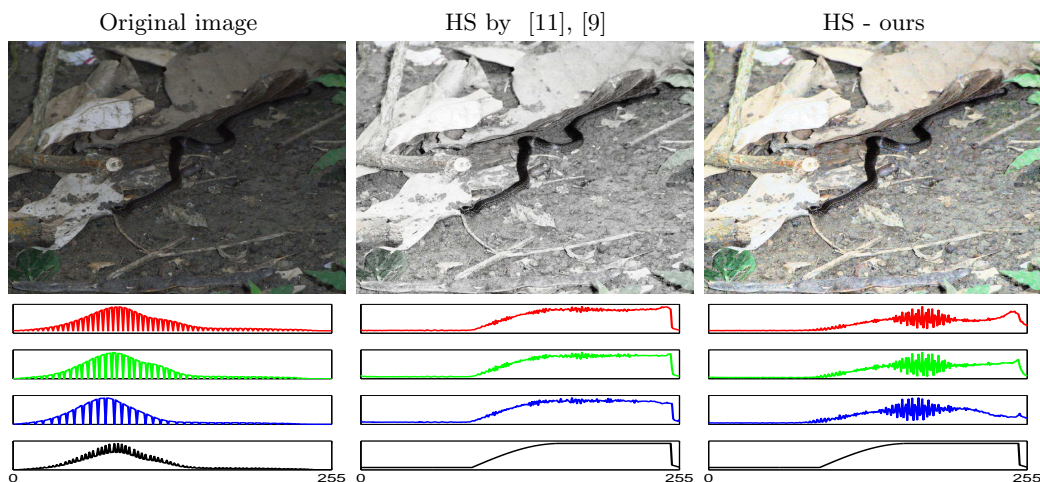

Fig. 4: Images and their histograms—R (■), G (■), B (■), luminance (■).

In the original image in Fig. 4 ($1000 \times 1000 \times 3$) there is a snake that is not easy to distinguish from the surrounding landscape. Our goal was to modify the histogram so that the snake is clearly seen. This is the reason why we chose as target histogram the curve on the bottom row of the restored images. For our algorithm, only 3.3% of the pixels were reprocessed by step 2.

5   Conclusions and perspectives

The sorting algorithm proposed in section 3 is for the present the best one. The proposed algorithm for color assignment in section 4 is fast and yields better results than the one used in [9]. However it does not exploit color perceptual facts that were used e. g. in [14]—but with an intensive computational cost. This point deserves further exploration.

References

1. T. Arici and S. Dikbas, *A histogram modification framework and its application for image contrast enhancement*, IEEE Trans. on Image Proc., 18 (2009), pp. 1921–1935.

2. N. Bassiou and C. Kotropoulos, *Color image histogram equalization by absolute discounting back-off*, Computer Vision and Image Understanding, 107 (2007).
3. F. Bauss, M. Nikolova, and G. Steidl, *Fully smoothed ℓ_1-TV models: Bounds for the minimizers and parameter choice*, J. of Math. Imaging and Vision, (2013), online.
4. R. S. Berns, *Billmeyer and Saltzman Principles of Color Technology*, Willey & Sons, Roy S., Ca, 3 ed., 2000.
5. V. Caselles, J. L. Lisani, J. M. Morel, and G. Sapiro, *Shape preserving local histogram modification*, IEEE Trans. on Image Processing, 8 (1999), pp. 220–229.
6. D. Coltuc, P. Bolon, and J.-M. Chassery, *Exact histogram specification*, IEEE Trans. on Image Processing, 15 (2006), pp. 1143–1152.
7. R. Gonzalez and R. Woods, *Digital Image Processing*, Addison-Wesley, 1993.
8. E. L. Hall, *Almost uniform distributions for computer image enhancement*, IEEE Transactions on Computers, C-23 (1974), pp. 207–208.
9. J. H. Han, S. Yang, and B. U. Lee, *A novel 3-D color histogram equalization method with uniform 1-D gray scale histogram*, IEEE Trans. on Image Proc., 20 (2011), pp. 506–512.
10. L. Menotti, L. Najman, A. d. Araújo, and J. Facon, *A fast hue-preserving histogram equalization method for color image enhancement using a bayesian framework*, in Proc. 14th Int. Workshop Syst., Signal Image Process, 2007, pp. 414–417.
11. S. F. Naik and C. A. Murthy, *Hue-preserving color image enhancement without gamut problem*, IEEE Trans. on Image Processing, 12 (2003), pp. 1591–1598.
12. M. Nikolova, Y. Wen, and R. Chan, *Exact histogram specification for digital images using a variational approach*, J. of Mathematical Imaging and Vision, (2012), online.
13. J. Ortega and W. Rheinboldt, *Iterative Solution of Nonlinear Equations in Several Variables*, Academic Press, New York, 1970.
14. R. Palma-Amestoy, E. Provenzi, M. Bertalmio, and V. Caselles, *A perceptually inspired variational framework for color enhancement*, IEEE Trans. on Pattern Analysis and Machine Intelligence, 31 (2009), pp. 458–474.
15. L. Schwartz, *Analyse: Topologoe générale et analyse fonctionnelle*, Hermann, Paris, 1993.
16. D. Sen and P. Sankar, *Automatic exact histogram specification for contrast enhancement and visual system based quantitative evaluation*, IEEE Trans. on Image Processing, 20 (2011), pp. 1211–1220.
17. C. Solomon and T. Breckon, *Fundamentals of Digital Image Processing. A Practical Approach with Examples in Matlab*, John Wiley & Sons, UK, 1 ed., 2011.
18. P. E. Trahanias and A. Venetsanopoulos, *Color image enhancement through 3-D histogram equalization*, in Proc. 15th Int. Conf. Pattern Recognit, vol. 1, 1997, pp. 545–548.
19. Y. Wan and D. Shi, *Joint exact histogram specification and image enhancement through the wavelet transform*, IEEE Trans. on Image Processing, 16 (2007), pp. 2245–2250.