Hue Constrained Image Colorization in the RGB Space

Fabien Pierre, Jean-François Aujol, Aurélie Bugeau, Vinh-Thong Ta

Noname manuscript No.
(will be inserted by the editor)

Hue Constrained Image Colorization in the RGB Space

Fabien Pierre · Jean-François Aujol · Aurélie Bugeau · Vinh-Thong Ta

Received: date / Accepted: date

Abstract This work proposes a novel variational framework for exemplar-based image colorization in RGB while most of existing methods use a luminance-chrominance space. Using directly these three RGB channels, our model does not need any pre or post-processing step. We design a primal-dual like algorithm that solves the proposed variational approach. The constraint when colorizing a gray-scale image is to maintain its luminance. In our RGB model this constraint is added in the variational model. Its preservation requires introducing a novel projection scheme that takes care of the hue value (H) of the HSI color-space. The behavior of our algorithm is illustrated on numerous examples, and a qualitative comparison is made with state-of-the-art methods.

Keywords Colorization · Total variation · Patches · Optimization

Fabien Pierre[1,2,3,4]
E-mail: fabien.pierre@math.u-bordeaux1.fr

Jean-François Aujol[1,2]
E-mail: Jean-Francois.Aujol@math.u-bordeaux1.fr

Aurélie Bugeau[3,4]
E-mail: aurelie.bugeau@labri.fr

Vinh-Thong Ta[4,5]
E-mail: vinh-thong.ta@labri.fr

[1]Univ. Bordeaux, IMB, UMR 5251, F-33400 Talence, France.
[2]CNRS, IMB, UMR 5251, F-33400 Talence, France.
[3]Univ. Bordeaux, LaBRI, UMR 5800, F-33400 Talence, France.
[4]CNRS, LaBRI, UMR 5800, F-33400 Talence, France.
[5]IPB, LaBRI, UMR 5800, F-33600 Pessac, France.

1 Introduction

Image colorization consists in turning a gray-scale image into a color one by adding some color information for each pixels. This treatment is used in the film industry to make old productions more attractive to the consumers, or to produce some special effects. It can also be used in embedded applications for sensor fusion [34]. The colorization problem can be explained as follows. The gray-scale image is considered to be the luminance channel. In order to preserve the initial image content, colorization methods impose that the luminance channel remains constant. The colorization problem can be reformulated as the problem of reconstructing the two chrominance channels. Combined with the luminance channel, that is the initial gray-scale image, a final RGB image is recovered. Most colorization systems therefore work in luminance-chrominance spaces, typically YUV or l$\alpha\beta$, only dealing with the chrominance channels without coupling them correctly with the luminance channel. As we will see later in this paper, this lack of coupling leads to some errors on the colorized images.

While turning a color image into a gray-scale one is only a matter of standard, the reverse operation is a strongly ill-posed problem. No information on which color has to be added is known. To add color information, all colorization systems need some prior information. In the state-of-the-art methods there exist two types of approaches. The first category includes the manual methods. The user puts some colors (referred as scribbles) directly onto the gray-scale image. He decides the position on the image and the exact color he wants, depending on the semantic and his own experience. A lot of methods have been proposed to solve the colorization problem using this prior [19, 22, 29, 30, 33].

(a) Source image. (b) Target image. (c) Our result.

Fig. 1 Exemplar-based image colorization. The source image (a) is used to obtain the colorized image (c) from the target (b).

For instance, the method of Levin *et al.* [22] diffuses the color information of the scribbles to the rest of the image with the assumption that chrominances have small variations if the luminance has small ones. The advantage of these methods is the regularity of the result. Among these methods the ones of [19] and [30] work in the RGB color-space. For each pixel without color neighbour of a colored pixel, the algorithm computes the nearest color (for the l^2 norm) to the ones of this neighbour. This color is chosen among all of it which luminance is equal to the gray level of the current pixel. However, all these manual methods have the same strong drawback: if the image represents a complex scene, the user has to put a lot of scribbles which is a tedious work.

The second category replaces the manual intervention by providing a color image (also called *source*) as prior information. This color image is used to colorize the gray-scale image (also called *target*). These approaches are called exemplar-based methods. In this paper, the proposed method belongs to the second category. Fig. 1 shows an exemplar-based colorization, where the result (Fig. 1(c)) is obtained with the proposed method.

One of the first exemplar-based methods has been proposed by Welsh *et al.* [32] and it is widely inspired by the algorithms for texture synthesis [14, 31]. This method works with the $l\alpha\beta$ color-space that provides decorrelated channels. For each pixel of the target, the method searches within the source for the pixel having the closest luminance and standard deviation in a patch of size 5×5. Finally, the α and β channels of this pixel are extracted and their values are attributed to the current pixel. Di Blasi *et al.* [13] and Chen *et al.* [9] improve this method. Di Blasi *et al.* propose to speed it up by tree clustering, and Chen *et al.* use an image matting to improve the regularity of the colorized image.

To the best of our knowledge, the first approach proposing exemplar-based prior with regularization of the colorized image is the method of Irony *et al.* [20]. First the image is segmented and a specified classifier based on DCT (discrete cosine transform) is used to initialize micro-scribbles on the gray-scale image. Finally, colors are diffused using the approach of [22]. Based on the same idea, Gupta *et al.* [17] propose to use more complex features (Gabor, SURF, intensity and standard deviation) computed from blocks of pixels (super-pixels, see [23] for details). The features are employed to provide a correspondence between textures of super-pixels in the target and color of the super-pixels in the source. The method places micro-scribbles on the gray-scale image at the center of each super-pixel and uses the approach of [22] to colorize the image. Authors of [17] remark that the use of super-pixel representation can be inaccurate for object boundaries or thin image structures (see results on Fig. 16 for instance).

The exemplar-based approach of Charpiat *et al.* [8] ensures a spatial coherency without a segmentation of the image but by minimizing the Euclidean norm between the 8 neighbors. The optimization of this problem is performed in two steps. First, the graph-cut algorithm is used, and then, a refinement is provided by a gradient descent. To make the link between textures and colors, SURF descriptors [2] are used. For the sake of simplicity, the authors prefer to avoid the colorization of the borders of the image, which is a practical drawback.

The method proposed by Bugeau *et al.* [5], provides a framework for exemplar-based colorization with a regularization by total variation (TV). The YUV color space is used. The initial gray-scale image is considered to correspond to the luminance channel Y, defined for a RGB color (R, G, B), as

$$Y(R, G, B) = 0.299R + 0.587G + 0.114B. \quad (1)$$

For each pixel x of the target, 8 chrominance candidates, denoted as c_i with $i = 1, \ldots, 8$, are extracted according to eight textural criteria computed on image patches. The choice of the best candidate is then made by minimizing a functional with a primal-dual like algorithm. Since the human vision is sensitive to small changes of colors when they are spatially close, the authors enforce the result to be regular. Hence, the model of Bugeau *et al.* [5], is given by the following functional, where $u = (U, V)$ is the chrominance to compute, and

Ω the image domain:

$$\begin{aligned} F_1(u,W) := TV(u)+ \\ \frac{\lambda}{2}\int_\Omega \sum_{i=1}^{8} w_i(\mathbf{x})\,\|u(\mathbf{x})-c_i(\mathbf{x})\|_2^2\;d\mathbf{x}+ \\ \alpha\int_\Omega \sum_{i=1}^{8} w_i(\mathbf{x})(1-w_i(\mathbf{x}))\,d\mathbf{x}+ \\ \chi_{U\in[0,255]^2}+\chi_{W\in\Delta}\ , \end{aligned} \tag{2}$$

where

$$TV(u)=\int_\Omega \sqrt{\sum_{K=U,V}\partial_x K(\mathbf{x})^2+\partial_y K(\mathbf{x})^2}\,d\mathbf{x}\ , \tag{3}$$

$$\begin{aligned} \Delta=\{(w_1,\dots,w_k) \text{ such that } \forall 1\le i\le k\ , \\ 0\le w_i\le 1 \text{ and } \textstyle\sum_i w_i=1\}\ , \end{aligned} \tag{4}$$

and

$$W:\mathbf{x}\mapsto\{w_i(\mathbf{x})\ ,\ i\in\{1,\dots,8\}\}. \tag{5}$$

The first term is the total variation (TV) used to regularize the image while allowing the image to have some discontinuities by preserving strong contours [28]. The term $\int_\Omega \sum_{i=1}^{8} w_i(\mathbf{x})\|u(\mathbf{x})-c_i(\mathbf{x})\|_2^2$ makes the link between the candidate color c_i and the selected u color. The weights w_i are used to select one candidate. To avoid mixing the colors, they are forced to go to 0 or 1 during the minimization process. Finally the authors make sure to retrieve color values in the range of $[0,255]^2$. Although this functional is not convex, it is minimized with a primal-dual algorithm. This method provides good results but the colorized images are too drab and contours are not preserved as illustrated in Fig. 2. These effects are due to the lack of coupling between YUV channels. The classical total variation on chrominance channels is not able to preserve contours.

Contributions This paper presents in details a new model for exemplar-based colorization working directly in the RGB color-space. This new model is based on the minimization of a functional inspired by [5]. Most of the state-of-the-art methods work in luminance-chrominance color-spaces. Since the proposed approach works in the RGB color-space, we do not convert the images into a luminance-chrominance space.

The proposed method is able to couple the different RGB channels of color which enables to get clean contours on the final results and shinier color images than the one of [5]. Hence, the new method does not require any post-processing step. The improvements come from the change of the color-space, the use of a coupled total variation (TV) and the suppression of a non-convex term of the functional.

The optimization on the RGB color-space of the new functional is difficult due to the numerous terms and the use of TV. Regularization by minimization of TV has been very useful in the image processing community since the work of [28]. In this paper, we describe the associated algorithm that solves these problems. It is inspired by the one of Chambolle and Pock [7] which uses a primal-dual formulation of the TV.

Even if we work in the RGB color space, we must ensure the full conservation of the luminance Y. Projections onto the luminance space are therefore included on the algorithm. We notice that the orthogonal projection is not relevant in the RGB color space for the colorization problem and introduce a new color projection which is able to preserve the hue (H channel of the HSI color-space) during the minimization process. The computation of this projection is detailed and a theoretical justification is provided. The colors of the final result are more realistic with this control of the hue.

[26] proposes a variational model working in the RGB color-space. This method does not ensure the preservation of the luminance, leading to blur effects.

Outline The paper is organized as follows: first, the new model working in the RGB color-space is presented. Next, a primal dual-like algorithm minimizing this model is given. We recall definitions and properties about the cylindrical color spaces (*e.g.*, HSI). In addition we explain how to compute the oblique projection in order to control the hue during the minimization process. Since non classical total variations for color images have been proposed in [15] and [3], a qualitative comparison with [15] is also provided. Finally, the convergence is numerically verified on numerous examples. We also study the behavior of this new algorithm on numerous examples and compare it with state-of-the-art colorization methods.

2 RGB Model for Colorization

The model of Bugeau *et al.* [5] produces good results. Nevertheless, images are too drab and halo effects are visible near strong contours due to a lack of coupling between the YUV channels. This coupling constitutes the major improvement produced by our method.

Another problem appears when the method retains a color candidate. Since the value of the luminance is given by the target image, the method only retains the chrominances. Changing the luminance of the color candidate while holding its chrominances constant gives

(a) Source image. (b) Target image. (c) Result with [5].

Fig. 2 Exemplar-based colorization results performed with strong regularization by the method of [5].

colors with various hue. Fig. 3 illustrates this effect: for a given U and V, different colors can be obtained when Y varies, leading to inconsistent hue. To solve this problem, we propose to work directly in the RGB color-space. Moreover, working in the RGB color-space avoids color transformation at input and output of the method as done in most existing approaches.

Fig. 3 If U and V are constant, and Y varies, different colors are obtained.

2.1 Presentation of the RGB Model

As in [5], assume that C different RGB candidates have been extracted for each pixel according to criteria based on texture features. In order to compute a regular image, the following functional is introduced, where $u = (R, G, B)$ stands for a RGB image:

$$\begin{aligned} F_2(u, W) := TV_{RGB}(u) + \\ \frac{1}{2\lambda} \int_\Omega \sum_{i=1}^{C} w_i(\mathbf{x}) \|u(\mathbf{x}) - c_i(\mathbf{x})\|_2^2 \, d\mathbf{x} + \\ \chi_{u \in [0,255]^3} + \chi_{Y(u)=I_g} + \chi_{W \in \Delta}. \end{aligned} \tag{6}$$

where

$$TV_{RGB}(u) = \int_\Omega \sqrt{\sum_{K=R,G,B} \partial_x K(\mathbf{x})^2 + \partial_y K(\mathbf{x})^2} \, d\mathbf{x}. \tag{7}$$

Other notations in (6) are the same as in (2).

In this model, the non-convex term $\alpha \int_\Omega \sum_{i=1}^{C} w_i(1 - w_i)$ of the chrominance model (2) is suppressed. Indeed, assume that u is constant and consider the term

$$\frac{1}{2\lambda} \int_\Omega \sum_{i=1}^{C} w_i(\mathbf{x}) \|u(\mathbf{x}) - c_i(\mathbf{x})\|_2^2 \, d\mathbf{x} + \chi_{W \in \Delta}. \tag{8}$$

The minimum of this term with respect to the variable W is given by

$$W = (0, \dots, 0, 1, 0, \dots, 0) \tag{9}$$

with 1 at the position i such that $\|u - c_i\| = \min_j \|u - c_j\|$. Intuitively, if u is constant, the minimum of (8) is realized when the closest candidate of u is used as data term. Without this non-convex term, the regularization of u has an influence on W. Moreover, the model becomes simpler and the new functional is convex with respect to u and W separately.

The constraint $Y(u) = I_g$ means that the luminance of the colorized image will be the same as the target one. This constraint preserves the initial target image content. The luminance constraint is obtained solving $A.u = I_g$ where u is the vector of three channels, R, G, B and I_g is the luminance of the target image, leading to $A = (0.299, 0.587, 0.114)$ thanks to equation (1).

2.2 Minimization of the RGB Model

Many optimization techniques dedicated to image processing have been developed. If the functional has only two terms, the classical forward-backward algorithm is sufficient [11, 12]. When there are more terms, the generalized forward-backward [27] is more appropriate. However, these methods require the computation of the proximal operator of the total variation which is computationally expensive. To overcome these limitations, a primal-dual formulation is proposed and solved by [7], avoiding inner loops in the case of TV minimization. To this end, the following dual problem is defined:

$$\begin{aligned} \min_u \max_p \; - \langle \operatorname{div} p | u \rangle + \frac{1}{2\lambda} \int_\Omega \sum_{i=1}^{C} w_i \|u - c_i\|_2^2 + \\ \chi_{u \in [0,255]^3} + \chi_{Y(u)=I_g} + \\ \chi_{B(0,1)}(p) + \chi_{W \in \Delta}, \cdot \end{aligned} \tag{10}$$

where $B(0, 1)$ is the unit ball of $\mathbb{R}^6$.

Following the ideas of [7], Algorithm 1 presents the different steps of the resolution scheme of the problem (10).

Algorithm 1 Primal-dual like algorithm minimizing the RGB model (6).

$Z_0 \leftarrow 0$, $W_0 \leftarrow 1/C$, $u_0 \leftarrow \sum_i w_i c_i$
for $n \geq 0$ **do**
  $Z_{n+1} \leftarrow P_B\left(Z_n + \sigma \nabla \overline{u}_n\right)$
  $W_{n+1} \leftarrow P_\Delta\left(W_n - \rho \frac{1}{\lambda}(\|\overline{u}_n - c_i\|^2)_i\right)$
  $u_{n+1} \leftarrow P_G\left(\frac{u_n + \tau\left(\mathrm{div}(Z_{n+1}) + \frac{1}{\lambda}\sum_i (w_n)_i c_i\right)}{1 + \frac{\tau}{\lambda}}\right)$
  $\overline{u}_{n+1} \leftarrow 2u_{n+1} - u_n$
end for

P_Δ is the projection onto the simplex Δ and is computed using [10]. $(\|u - c_i\|^2)_i$ represents the array of the same size of W such that each weight is equal to $\|u(\mathbf{x}) - c_i(\mathbf{x})\|^2$ for each index $i \in 1, \ldots, C$ and for all pixels $\mathbf{x} \in \Omega$.

P_B represents the projection onto the unit ball of $\mathbb{R}^6$.

P_G is the projection onto L_{I_g}, which stands for the intersection:

$$L_{I_g} := [0, 255]^3 \cap \{u \text{ such that } Y(u) = I_g\}. \tag{11}$$

We use the following definition for $u : \mathbb{R}^2 \to \mathbb{R}^3$:

$$u(x, y) = (R(x, y), G(x, y), B(x, y)) := (u_{x,y,1}, u_{x,y,2}, u_{x,y,3}). \tag{12}$$

The discrete gradient ∇ and divergence div operators are defined as in [4]. For the sake of clarity, the following definitions only consider square images of size $N \times N$ but extension to rectangular case is straightforward.

Definition 1 Let

$$\nabla u_{x,y} = \begin{pmatrix} \begin{cases} u_{x+1,y,1} - u_{x,y,1} \text{ if } x < N, \\ 0 \text{ if } x = N, \end{cases} \\ \begin{cases} u_{x,y+1,1} - u_{x,y,1} \text{ if } y < N, \\ 0 \text{ if } y = N. \end{cases} \\ \vdots \\ \begin{cases} u_{x,y+1,3} - u_{x,y,3} \text{ if } y < N, \\ 0 \text{ if } y = N. \end{cases} \end{pmatrix}. \tag{13}$$

Next, we propose the dual operator:

$$\begin{aligned} \mathrm{div}(p)_{x,y} = & \begin{cases} p^1_{x,y} - p^1_{x-1,y} \text{ if } 1 < x < N, \\ p^1_{x,y} \text{ if } x = 1, \\ -p^1_{x-1,y} \text{ if } x = N, \end{cases} \\ & + \ldots \\ & + \begin{cases} p^6_{x,y} - p^6_{x,y-1} \text{ if } 1 < y < N, \\ p^6_{x,y} \text{ if } y = 1, \\ -p^6_{x,y-1} \text{ if } y = N, \end{cases} . \end{aligned} \tag{14}$$

In the convex case the behavior of primal dual algorithms has been studied in [7]. For one channel, the square of the operator norm of the divergence, neglecting borders effect, is equal to 8 (see [6], Remark page 92). The convergence of this algorithm to a minimizer is verified for $\tau, \sigma > 0$ such that $\tau\sigma < 1/8$ (see [7], Theorem 1). In the case of three color channels, neglecting borders effects, and simplifying to a square image:

$$\begin{aligned} \|\mathrm{div}\, p\|_2^2 &= \sum_{1 \leq x,y \leq N} \left(p^1_{x,y} - p^1_{x-1,y} + \ldots - p^6_{x,y-1}\right)^2 \\ &\leq 12 \sum_{1 \leq x,y \leq N} \left(p^1_{x,y}\right)^2 + \ldots + \left(p^6_{x,y-1}\right)^2 \\ &\leq 24 \|p\|_2^2 \end{aligned} \tag{15}$$

Choosing $p^1_{x,y} = p^2_{x,y} = \ldots = p^6_{x,y} = (-1)^{x+y}$, we obtain that

$$\kappa = |||\,\mathrm{div}\,||| = \sup_{\|p\| \leq 1} \|\mathrm{div}\, p\|_2\,, \tag{16}$$

the norm of the divergence operator can be approximated with $\kappa = 24 - O(M)$, with $M = \text{width} + \text{height}$ in pixels of the considered image. The value 24 is retained for the square of the norm of the operator div, equal to the gradient operator one (Theorem IV.5.6 in [21]).

Since the functional is convex with respect to u, our algorithm computes the solution of the saddle-point problem (10) with respect to this variable. To that end, the proximal operator of the constraint $\chi_{u \in [0,255]^3} + \chi_{Y(u) = I_g}$ must be computed. It is equal to 0 on the intersection between a plan and the cube $[0, 255]^3$ and equal to $+\infty$ otherwise. To satisfy this constraint the projection is computed, first on the plan, and second on the intersection of the plan and the cube (see Appendix A).

With the implementation of this projection, the primal-dual algorithm, minimizing the functional (6) with respect to u and W is summarized in Algorithm 1. In

the iteration of the primal-dual algorithm, a step of gradient descent is added with respect to W.

At this point, the algorithm does not produce satisfying results (see Fig. 7). The contours are well respected and the images are not drab, but they are now too shiny, containing irrelevant colors due the orthogonal projection P_G that does not preserve the hue. The following section explains how to compute suitable colors.

3 Hue Consistent Projection

The algorithm presented in the previous section suffers from a lack of consistency of the hue when computing the orthogonal projection P_G. This is due to unequal weights in the definition of the luminance (1). We solve this problem by replacing this projection by an oblique one. In the following, we review the HSI color-space that enables to propose a hue consistent oblique projection.

3.1 A Geometrical Point of View of the HSI Color-space

The RGB color-space is not adapted to the human description of colors. Indeed, the description of colors with the three values of primaries of the RGB color-space is not intuitive. People prefers to describe color with words such as *red*, *purple* or *orange*. To that end, particular color-spaces have been created. For instance, the HSV color-space defines colors with three particular channels which are the hue, the saturation and the value. In the same trend, other spaces exist, *e.g.*, HSI, HSL or HSY. In the following, we focus on the HSI color-space (presented in [16]) for its simplicity in term of geometric interpretation. This point of view has been used recently for image enhancement by Nikolova *et al.* [24].

The hue represents the human perception of the pure color. The hue is not defined if $R = G = B$, *i.e.*, the color *gray* has no hue. Otherwise, the hue H is defined as

$$H = \begin{cases} \theta & \text{if } B \leq G \\ 360 - \theta & \text{otherwise,} \end{cases} \tag{17}$$

where

$$\theta = \arccos \left\{ \frac{\frac{1}{2}[(R-G)+(R-B)]}{\sqrt{(R-G)^2+(R-G)(G-B)}} \right\}, \tag{18}$$

which is an angle in degrees. All the points in the RGB color-space which have the same hue are on the same open half-plane. The edge of this half-plane is the axis passing through coordinates $(0, 0, 0)$ (black b) and (255, 255, 255) (white w). For a given color u, the plan of all the colors with the same hue as u contains u, b and w (Fig. 4). The only operation letting H constant is addition with the combination of the two vectors $\overrightarrow{bu}$ and $\overrightarrow{bw}$. Indeed, these two vectors constitute a basis of the plan.

The human perception uses two other descriptors: the saturation S and the intensity I. The saturation represents how the color is mixed with the white:

$$S = 1 - \frac{3}{R+G+B}[\min(R, G, B)]. \tag{19}$$

It is equal to 0 if the color is gray, and has its maximum value if the color is away from the axis (bw) (Fig. 4).

The value of the intensity I can be compared to the illumination of an electric light. If the light is turned off, the intensity is low. If it is turned on, the value of intensity is high. I is defined as:

$$I = \frac{R+G+B}{3}. \tag{20}$$

The parametrization of $\mathbb{R}^3$ by the coordinates HSI describes all this space.

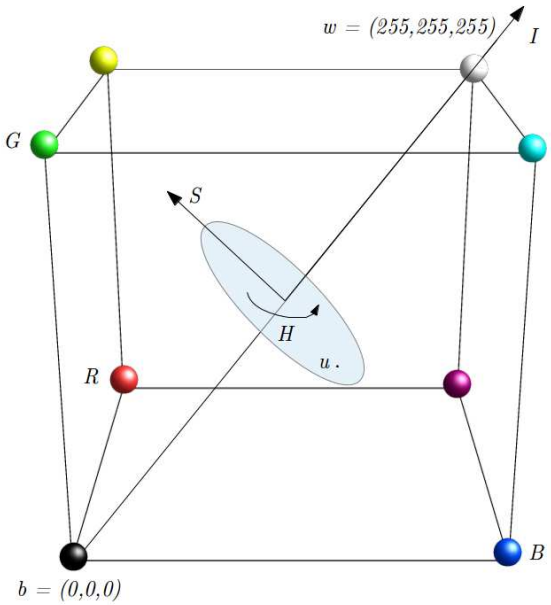


Fig. 4 The HSI components represented in the RGB color-space. All the RGB colors which have the same hue as u are on the half-plane containing u and the line (bw).

To manipulate the components on an image by modifying only one or two values of the HSI color-space, a simple method consists in first transforming the RGB

color into HSI, then work on these new values, and finally invert the transformation. Nevertheless, the new RGB values are not guaranteed to be inside the cube $[0, 255]^3$, *i.e.*, the new value might not respect the standard range of an image. It is therefore preferable to work directly in the RGB color-space. For instance, if the H channel must remain constant during an operation on a color, one must ensure that the new color value stays in the open half-plane containing w, b and the original value.

At each iteration of our colorization process, we are obtaining a RGB color image $\tilde{u}$. Given the luminance l of the original image, we are facing the problem of finding a RGB vector u having the same hue as $\tilde{u}$. It is solved by projecting $\tilde{u}$ on the half-line

$$\{u \text{ s.t. } H(u) = H(\tilde{u})\} \cap \{u \text{ s.t. } Y(u) = l\}$$

staying in the half-plane $\{u \text{ s.t. } H(u) = H(\tilde{u})\}$. To avoid changing the hue, it is important to stay in the open half-plane, but the orthogonal projection on the line $Y(u) = l$ can produce a color on the opposite half-plane (see Fig. 5). To overcome this limitation, we propose in this work to use a non-Euclidean projection in the direction of the I axis. A similar problem (preservation of the range and the hue) has been addressed in [24] for image enhancement. The main advantage of our approach compared to [24] is its simpler geometric interpretation.

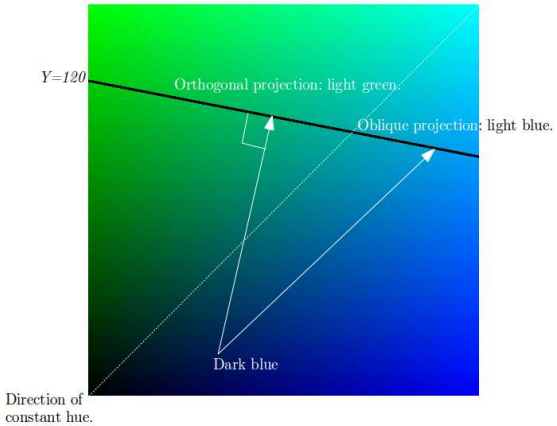


Fig. 5 The orthogonal projection of a color (here dark blue) onto the constraint $\{u \in \mathbb{R}^3 \text{ such that } Y(u) = I_g\}$ changes the hue. The color is projected onto the light green instead of an other blue. The oblique projection permits to obtain a color respecting luminance condition and that has the same hue as the initial dark blue.

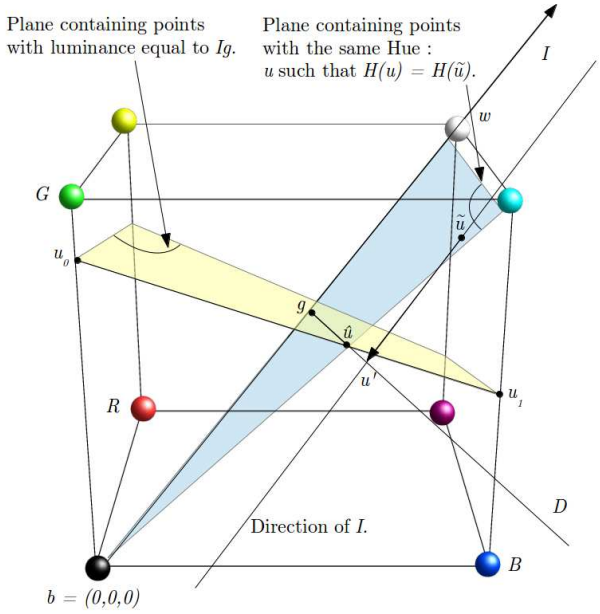


Fig. 6 The projection of $\tilde{u}$ on the convex set $L_{I_g} \cap \{H(\tilde{u}) = H(u)\}$ is computed in two steps. First the oblique projection of $\tilde{u}$ on $\{\hat{u} \text{ such that } Y(\hat{u}) = I_g\}$ in the direction of I is computed and denoted by u'. Secondly, the algorithm computes $\hat{u}$, the closest point of u' on the convex set $L_{I_g} \cap \{H(\tilde{u}) = H(u)\}$

3.2 An Oblique Projection

Due to the unequal weights in the definition of the luminance (1), the orthogonal projection P_G onto the plan $\{u \text{ such that } Y(u) = l\}$ changes the hue. For instance, a pure blue $(0, 0, 2)$ with $H = 0.67$ is mapped to $(66, 131, 27)$ which is a green with $H = 0.17$. In this section a new projection $\tilde{P_G}$, maintaining the hue constant, is introduced and formalized. Instead of projecting orthogonally a color $\tilde{u}$ on L_{I_g} (yellow plane in Fig. 6), the projection is chosen on the half-plane containing colors with the same hue as $\tilde{u}$ (blue plane in Fig. 6). More precisely, the point $\tilde{u}$ is projected onto the plane $\{u \text{ such that } Y(u) = l\}$ with direction $\overrightarrow{bw}$, giving a point denoted as u'. Along this axis the hue is constant, and this first result u' is a point with correct hue and luminance.

If this point u' is out of range, a solution with the same hue has to be computed. To this end, consider the half-line passing through u' and crossing the (bw) half-line (D on Fig. 6). D is contained in the plane $\{u \text{ such that } Y(u) = l\}$ (yellow plane on Fig. 6). On this half-line, the luminance and the hue are correct. Therefore, solutions in the RGB cube lie on D. The nearest of them u is retained as a final result, in order to avoid as much as possible to reduce the saturation (u on Fig. 6).

In the following, we detail the different steps of this new oblique projection. First, we need to compute the projection u' onto D corresponding to the intersection

$$\{H(\tilde{u}) = H(u)\} \cap \{Y(u) = I_g\} \tag{21}$$

in the direction of $\overrightarrow{bw}$. In the following, the coordinates of u in the RGB color-space are denoted by (R, G, B). To compute this projection, we define γ such that

$$Y\left[(R, G, B) + \gamma\,(1, 1, 1)\right] = I_g. \tag{22}$$

Thus, $\gamma = I_g - Y(R, G, B)$.

After this first result (denoted $u' = (R', G', B')$), the projection onto the polygon L_{I_g} of u' must be computed. u' is on the half line D. There are now two possibilities: u' is inside the cube $[0, 255]^3$ or not. If so, no further processing is needed. Otherwise the half line D is parametrized as follows: the point $g = (I_g, I_g, I_g)$ and u' are on D. The projection of u' is equal to $g + \beta\left((R', G', G') - g\right)$. The computation of β is now described. For each edge of the polygon L_{I_g} denoted by $[u_0, u_1]$, the following system is solved with respect to α and β:

$$\begin{cases} u = g + \alpha(u' - g) \\ u = u_0 + \beta(u_1 - u_0) \end{cases} \tag{23}$$

where u is the intersection of the line (u_0u_1) and the line (gu'). The detailed expression of α and β is given in Algorithm 2. If $\alpha \in [0, 1]$ and $\beta \in [0, 1]$ then the intersection of the half-line D and the edge $[u_0, u_1]$ is given by $g + \alpha(u' - g)$ which corresponds to the final result.

This oblique projection has a variational formulation described hereafter. Compared to the orthogonal projection, which is the proximal operator of a convex function:

$$P_G(\tilde{u}) = \text{argmin}_u \, \|u - \tilde{u}\|^2 + \chi_{u\in[0,255]^3} + \chi_{Y(u)=I_g}\ , \tag{24}$$

the oblique projection stands for:

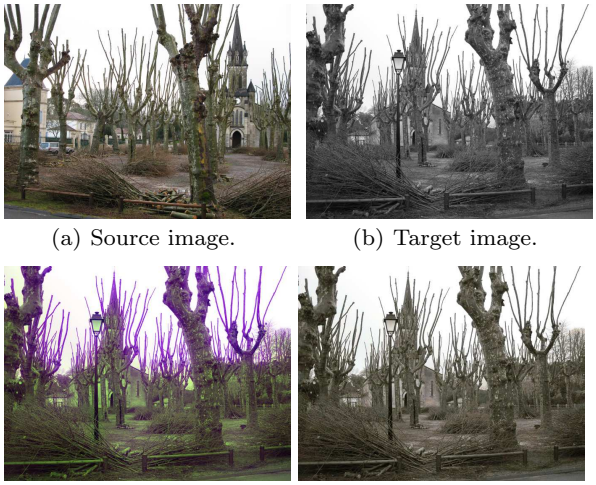
$$\tilde{P}_G(\tilde{u}) = \text{argmin}_u \, \|A(u - \tilde{u})\|^2 + \chi_{u\in[0,255]^3} + \chi_{Y(u)=I_g} + \chi_{H(u)=H(\tilde{u})}\ , \tag{25}$$

where A is an invertible matrix chosen such that $\text{argmin}_u \, \|A(u - \tilde{u})\|^2 + \chi_{\mathcal{C}}$ is the projection onto a vectorial subspace $\mathcal{C}$ in the direction of the line (bw). The existence of this matrix is detailed in Appendix B.

The complete algorithm solving (25) is given in Algorithm 2, and is justified in Appendix B. Finally, the orthogonal projection P_G in Algorithm 1 is replaced by $\tilde{P}_G$ computed with Algorithm 2.

Algorithm 2 Computation of the solution of (25).

$u = (R, G, B)$ $u' = (R', G', B')$ $u_i = (R_i, G_i, B_i)$
$\gamma \leftarrow 3(I_g - Y(u))$
$u' \leftarrow u + \dfrac{\gamma}{3}(1, 1, 1)$
$\Lambda \leftarrow \emptyset$
for $\{u_i, u_j\}$ points of the polygon L_{I_g} **do**
  $d \leftarrow (R' - I_g)(G_i - G_j) - (G' - I_g)(R_i - R_j)$
  if $d \neq 0$ **then**
    $\beta \leftarrow \dfrac{(R_i - b^x)(I_g - G') + (R' - I_g)(G_i - I_g)}{d}$
    if $R' - I_g \neq 0$ **then**
      $\alpha \leftarrow \dfrac{(R_i - I_g) + \beta(R_i - R_j)}{R' - I_g}$
    else if $B' - I_g \neq 0$ **then**
      $\alpha \leftarrow \dfrac{(B_i - I_g) + \beta(B_i - B_j)}{B' - I_g}$
    else
      projection $\leftarrow (I_g, I_g, I_g)$ is the final result.
    end if
    $\alpha \leftarrow \dfrac{1}{d}\left[(R_i - I_g)(G_i - G_j) - (G_i - I_g)(R_i - R_j)\right]$
    $\beta \leftarrow \dfrac{1}{d}\left[(I_g - R')(G_i - I_g) - (R' - I_g)(G_i - I_g)\right]$
    if $\alpha \in [0, 1]$ and $\beta \in [0, 1]$ **then**
      $\Lambda \leftarrow \Lambda \cup \alpha$
    end if
  end if
end for
$M_\alpha \leftarrow \max \Lambda$
projection $\leftarrow g + M_\alpha(u' - g)$



(a) Source image. (b) Target image.

(c) Colorization with orthogonal projection. (d) Colorization with oblique projection.

Fig. 7 Colorization results with the orthogonal and the oblique projections. The oblique projection is able to hold the hue constant and therefore produces better results than the orthogonal projection.

Fig. 7 illustrates the benefits of the oblique projection for image colorization, in particular when the

source image is drab (Fig. 7(a)). In this figure, we can see that in the case of the orthogonal projection, the result of the algorithm can produce some unrealistic colors (Fig. 7(c)). Indeed, the orthogonal projection onto L_{I_g} is not able to respect the original hue. With the oblique projection, the algorithm computes colors which are perceptually close to the source image (Fig. 7(d)).

4 Experimental Results

In this section, we present the results obtained with our method. We also provide a numerical analysis of our approach and a comparison with state-of-the-art methods.

4.1 Features and Candidates Extraction

In this section, we describe the process to extract color candidates. To extract these candidates, we follow the same scheme as in [5]. The first step consists in converting the source image into a gray-scale one using (1). In order to make this converted image comparable to the target one, a linear transformation [18] of the two grayscale images (luminance-remapping) is applied. Notice that, in order to speed up our approach, the search for the candidates is performed over a randomly subsampling of 200 pixels of the source image. [5] choose empirically the following three features based on image patches:

- standard deviation for patches of size 5×5 and 3×3
- the spectral amplitude of the Discrete Fourier Transform for patches of size 7×7, 9×9 and 11×11
- the cumulative histogram for patches of size 7×7, 9×9 and 11×11

leading to a set of 8 possible candidates for each pixel of the target image.

These criteria and the associated distances are detailed as follows. For a patch $P_{\mathbf{x}}$, centered on pixel $\mathbf{x}$, the first distance $\rho_1(\mathbf{x}, \mathbf{y}, P)$ between the pixels $\mathbf{x}$ (resp. $\mathbf{y}$) in the target (resp. source) image, is defined as:

$$\rho_1(\mathbf{x}, \mathbf{y}, P) := |\sigma^2(P_{\mathbf{x}}) - \sigma^2(P_{\mathbf{y}})| \ , \tag{26}$$

where $\sigma^2(P_{\mathbf{x}})$ (resp. $\sigma^2(P_{\mathbf{y}})$) represents the variance of the luminance values in $P_{\mathbf{x}}$ (resp. $P_{\mathbf{y}}$). For a given pixel $\mathbf{x}$ of the target image, the nearest pixel according to this distance is found in the source image (converted into a gray-scale image). The RGB values of this nearest pixel are kept for further processing.

Let $\hat{P}_{\mathbf{x}}(\xi)$ be the Fourier transform of the patch $P_{\mathbf{x}}$. The second distance is given by:

$$\rho_2(\mathbf{x}, \mathbf{y}, P) := \sum_{\xi} \left| |\hat{P}_{\mathbf{x}}(\xi)| - |\hat{P}_{\mathbf{y}}(\xi)| \right| \ , \tag{27}$$

and the last distance is defined as:

$$\rho_3(\mathbf{x}, \mathbf{y}, P) := \sum_{i} \left| H_{P_{\mathbf{x}}}(i) - H_{P_{\mathbf{y}}}(i) \right| \ , \tag{28}$$

where $H_{P_{\mathbf{x}}}$ (resp. $H_{P_{\mathbf{y}}}$) corresponds to the cumulative histogram of the patch $P_{\mathbf{x}}$ (resp. $P_{\mathbf{y}}$). This distance is equivalent to the Wasserstein distance. To extract the candidates, the method proceeds as with the first distance.

4.2 Parameter Settings

Influence of λ. Fig. 8 presents the influence of the parameter λ (6). If λ is low, the regularization has no influence on the final result and the colors are not realistic. The algorithm does not take care of the coupling of the contours, see for example on the right above the mountain edges. If the parameter λ is high, the resulting image is drab (see Fig. 8(c)).

We found that $\lambda = 5.10^3$ provides satisfying results, but a refinement of λ for each image can slightly improve the result. Other parameters of Algorithm 1 are fixed and are chosen as follow: $\rho = 1$, $\tau = 2$ and $\sigma = 0.01$.

(a) $\lambda = 10^2$.

(b) $\lambda = 10^3$. (c) $\lambda = 10^4$.

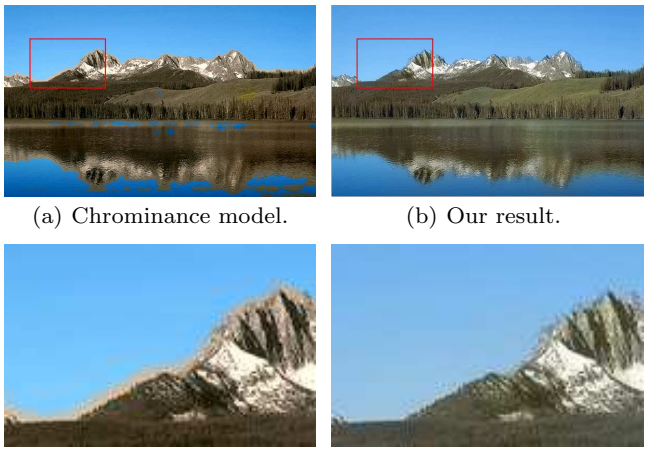
Fig. 8 Influence of the λ parameter on the result. If λ is high, the regularization of the image is strong, if it is low the regularization is weaker and the result is closer of the labelling $\sum_i w_i c_i$. With a strong regularization, the result is drab.

All the experiments presented in this paper are performed on a standard computer equipped with an Intel

Xeon Processor X5550 with 24 Go of RAM. The colorization of an image of size 680×512 pixels takes about $290sec$ on this equipment. This time is divided into two parts: $170sec$ for the search of candidates, and $120sec$ for the minimization of the functional (600 iterations). The search of candidates could easily be accelerated with recent methods such as [1].

4.3 Analysis of the Behavior of the Method

Influence of coupled TV. Fig. 9 compares the proposed method with the chrominance model of [5] and shows the influence of the coupled TV. With our RGB model and our oblique projection, images are less drab (Fig. 9(b)), and contours are well preserved (Fig. 9(d)) as compared to [5] (Fig.s 9(a) and 9(c)). This figure clearly demonstrates that the coupling of the RGB channels is efficient in order to preserve contours during colorization process as compared to the chrominance model of [5].

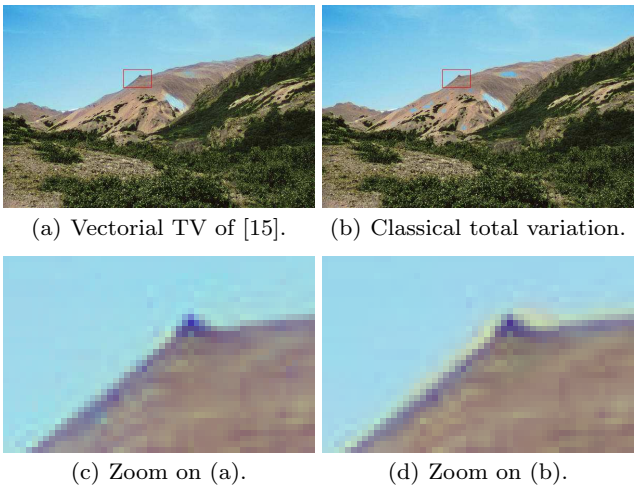


(a) Chrominance model. (b) Our result.

(c) Zoom on (a) (d) Zoom on (b)

Fig. 9 Comparison of different regularization. With the RGB model, the coupling is better than with the luminance-chrominance one of [5]. This better coupling produces contours without halo effect.

In the definition of the coupled total variation used in this paper, the three channels are only spatially coupled. Recently, the vectorial total variation introduced by Goldluecke *et al.* in [15] also couples contours directionally. With this regularization term, the minimization of the associated functional is performed by simply replacing P_B in Algorithm 1 with the projection described in detail in [15]. Fig. 10 shows the effects of this vectorial TV term. The colorized image looks better if the three color channels are directionally coupled. In the case of strong changes of direction on the contours, the consistency of color is improved as compared



(a) Vectorial TV of [15]. (b) Classical total variation.

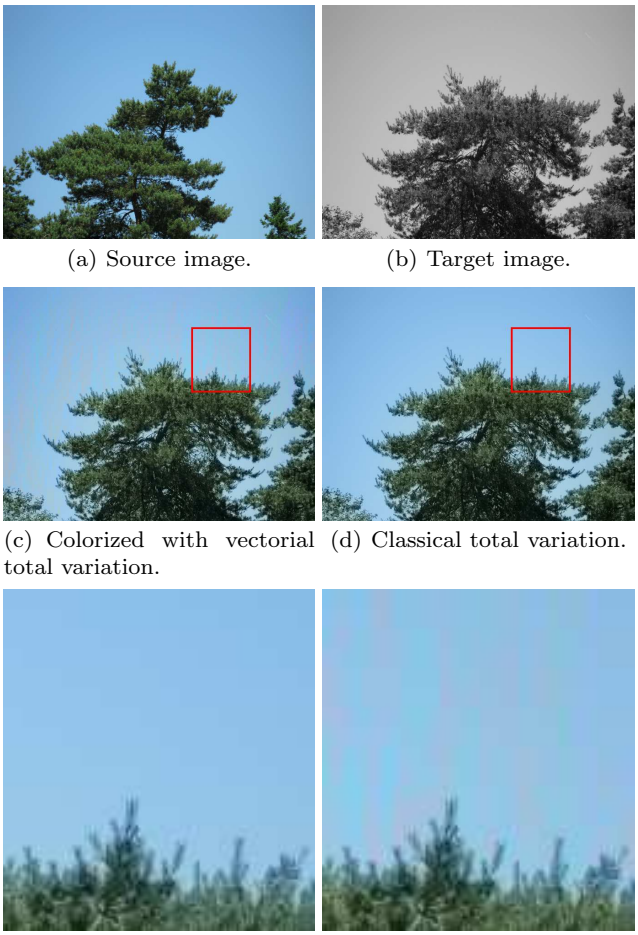
(c) Zoom on (a). (d) Zoom on (b).

Fig. 10 Comparison of the vectorial total variation proposed in [15] and the classical total variation defined in (7). On noisy contours with strong changes of direction, the vectorial total variation improves the results and avoids slight halo effect.

to the classical TV where some little halo effects are visible on Fig. 10(d). These halo effects disappear with the vectorial total variation. Nevertheless, this vectorial TV creates artifacts in case of complex contours and strong regularization (Fig. 11). In Fig. 11(f) the constant part (the sky) contains pink artifacts (Fig. 11(f)). This problem might be explained by the matching of the directions of the three color channels. Moreover, the minimization of this TV term is time consuming inducing a slower convergence than the classical TV due to a more complex projection. For all these reasons, we do not consider this vectorial total variation in this paper and we prefer to use the classical one.

Influence of the non-convex Term. In the new proposed functional, the non-convex term $\int_\Omega \sum_i w_i(1-w_i)$ [5] has been removed. We explain this choice in the following.

In order to study the behavior of the model, it is necessary to verify numerically the convergence of the algorithm. In Fig. 12(a) we can see that the value of the functional decreases and becomes asymptotically constant. This result demonstrates that our algorithm numerically converges. After convergence, Fig. 12(b) shows that only 1.8% of weights W do not go to 0 or 1. It corresponds to about 14.5% of the pixels of the image. These outliers do not come from the absence of the non-convex term but to the case where the metrics choose at least the two same color candidates. The algorithm is therefore unable to decide which candidate it has to choose. The random choice of 8 candidates over the set of 200 pixel gives a probability of 13.22% to have at least 2 similar candidates. The value of the histogram of W (Fig. 12(b)) is concentrated in 0 and 1,

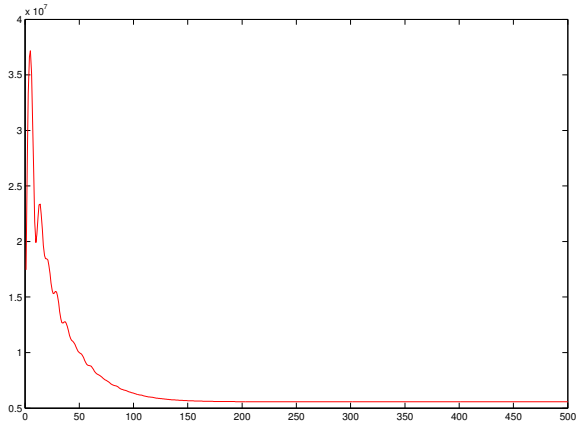


(a) Source image. (b) Target image.

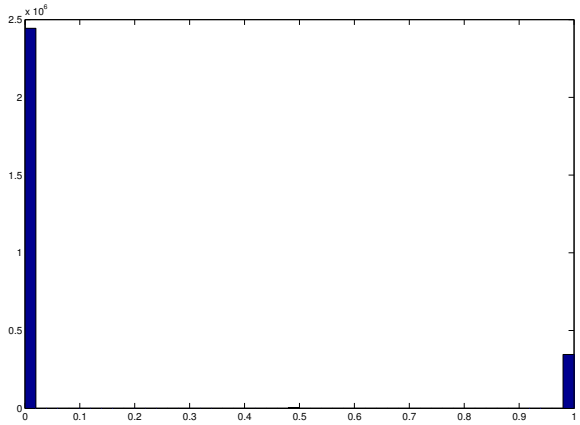
(c) Colorized with vectorial total variation. (d) Classical total variation.

(e) Zoom on 11(d). (f) Zoom on 11(c).

Fig. 11 Comparison of the vectorial total variation proposed in [15] and the classical total variation defined in (7) on images with complex contours. The vectorial total variation produces artifacts on the constant parts near the contour.



(a) The value of the functional is decreasing during the convergence of the algorithm.



(b) Histogram of W at convergence of the algorithm.

Fig. 12 Histogram of W and value of the functional during the iterations of the algorithm, for the colorization of 9(b). The value of the functional during the minimization process shows that the algorithm converges to a fix point. The histogram contains weights equal to 0 and 1, which provides a labeling without mixing of colors.

even without the term $\int_\Omega \sum_i w_i(1-w_i)$ that shows that it is unnecessary in our model, since the weights goes naturally to 0 and 1.

Fig. 13 shows the influence of the non-convex term, where Fig. 13(b) is the labeling provided with, and Fig. 13(c) without. The labeling is equal to $\sum_i w_i c_i$, i.e. to the minimum of the term $\int_\Omega \sum_i w_i \|u-c_i\|_2^2$ with respect to u. The initialization is presented in Fig. 13(a). With this non-convex term, the final value of W is not able to be modified and the result of the algorithm does not depend on the regularization. It only depends on the initialization. Without this non-convex term, the values of W can change if the value of u becomes closer to another color candidate. Thus the regularization of u has an influence on W. The consequence is a better regularity of the data-fidelity term $\sum_i w_i c_i$ which corresponds to the final labeling provided by the algorithm. Fig. 13 illustrates this effect. The data-fidelity term is more regular without the non-convex term (Fig. 13(c)). The suppression of this term improves the results.

4.4 Results and Comparisons

In this section, we provide colorization results obtained by our method and compare with state-of-the-art approaches.

Fig. 14 shows a comparison of our method with approaches of Irony *et. al.* [20], Welsh *et. al.* [32] and Gupta *et. al.* [17] where the results are taken directly from the paper of [17]. On the left, the targets and sources are shown. Our results are in the third column. The results of [20], and [8] are not so realistic although the algorithms are complex. The images of [32] are acceptable but the absence of regularization involves damaging artefacts. Indeed, areas that were originally ho-

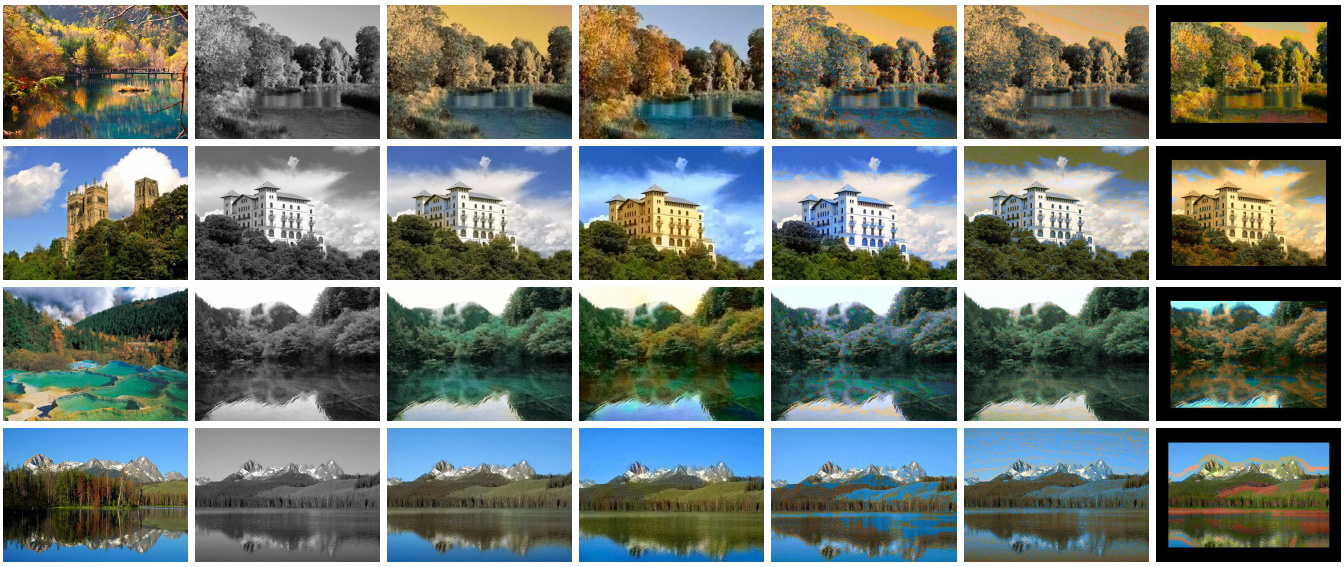


Fig. 14 Comparison with state-of-art methods. On the left, the target image in gray-scale and the source image in color. Results of our method on four images, compared with the results of the state-of-the-art methods presented in [17] ([17], [20], [32] and [8]). Our results are comparable to [17] but our algorithm works without any preliminary segmentation.

(a) Initialization.

(b) Labeling provided with the non-convex term.

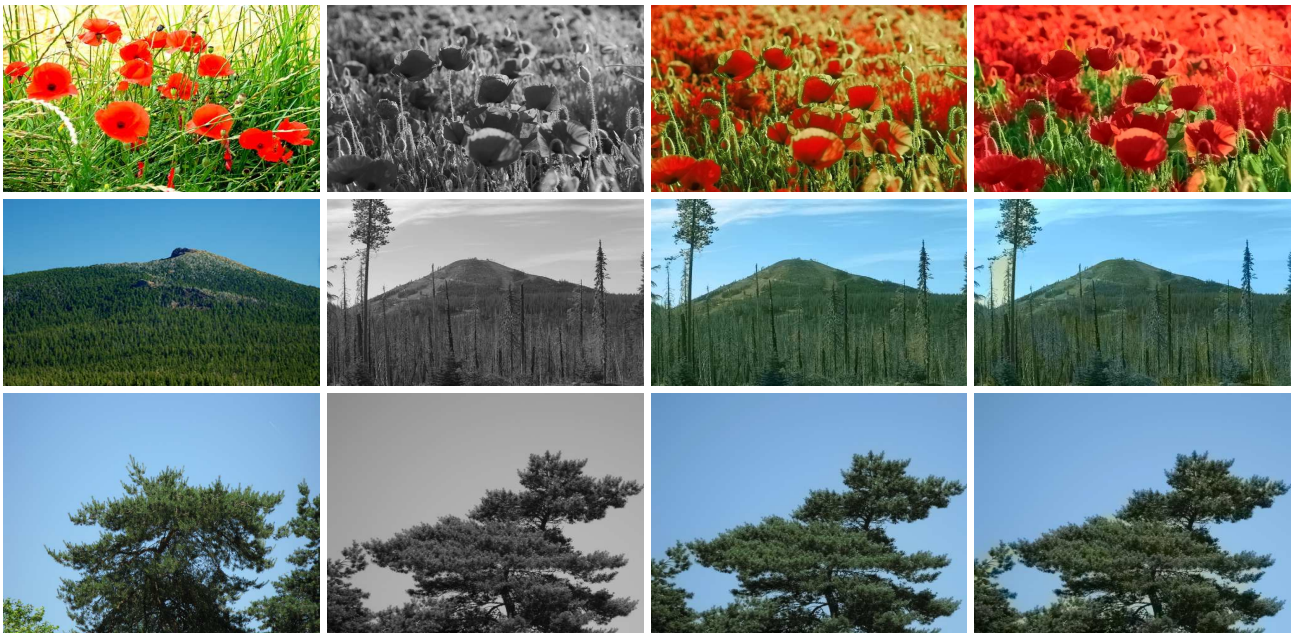
(c) Labeling provided without the non-convex term.

Fig. 13 Labeling provided by the algorithm: $\sum_i w_i c_i$ at convergence and at initialization. This term corresponds to the minimum of the functional with respect to u in the case $\lambda = 0$, with weights obtained at convergence of the algorithm.

Fig. 15 Results obtained with our method on different types of images.

mogeneous now present some irregularities. Our algorithm better preserves the homogeneous parts, as in the sky. The quality of our results are comparable to the one of Gupta *et al.* [17], whereas our algorithm is much simpler since our method does not require local segmentation like super-pixels [23]. Nevertheless, such segmentation can sometimes have a positive effect, which can be seen for instance in the sky of the first line of Fig. 14.

Fig. 16 reads comparisons of our result with ones performed with the method of [17]. Image of the first line shows that their texture features are not better that ours, and this method can fail. The second and the third lines show that the method of [17] fail on thin structures. For example, for the second line, at the left of the image, between trees the color is not consistent for the method of [17] contrary to our which provides a good color. At the third line, there is a gray color between leaves of the tree on the method of [17] instead of our.



Source. Target. Our result. [17].

Fig. 16 Comparison with Gupta for images with this structures. state-of-art methods. On the left, the target image in gray-scale and the source image in color. Results of our method on four images, compared with the results of the state-of-the-art methods presented in [17] ([17], [20], [32] and [8]). Our results are comparable to [17] but our algorithm works without segmentation.

Finally, Fig. 15 provides additional colorization results. This figure clearly demonstrates the efficiency and the potential of our method in order to colorize images of various types.

5 Conclusion and Future Work

In this paper a new variational model for image exemplar-based colorization has been presented. This variational model improves the chrominance model of [5] by working directly in the *RGB* color-space. An oblique projection respecting a given hue and a given luminance has been proposed to improve colorization results. A new primal-dual like algorithm has been described and its behavior numerically studied. In future works, we plan: to propose an interactive method that would solve the problems in the constant parts of the images, to increase the speed of convergence of the method by testing discrete algorithms, and to use the generalized total variation [3] on the problem of colorization.

Acknowledgments

This study has been carried out with financial support from the French State, managed by the French National Research Agency (ANR) in the frame of the Investments for the future Programme IdEx Bordeaux (ANR-10-IDEX-03-02). J-F. Aujol is a member of Institut Universitaire de France. The authors would like to thank Raj Kumar Gupta for providing the images presented in [17].

A About the Orthogonal Projection

The natural problem that arises when implementing Algorithm 1 is the projection of u onto its constraints, *i.e.*, the computation of the proximal operator of $\frac{\lambda}{2}\int_\Omega \sum_{i=1}^C w_i\|u-c_i\|_2^2+\chi_{u\in[0,255]^3}+\chi_{Y(u)=I_g}$. Indeed, defining

$$F(u)=\frac{1}{2\lambda}\int_\Omega \sum_{i=1}^C w_i\|u-c_i\|_2^2+\chi_{u\in[0,255]^3}+\chi_{Y(u)=I_g}, \quad (29)$$

$$\begin{aligned}&\text{prox}_F(\tilde{u})\\ &=\text{argmin}_u \|u-\tilde{u}\|_2^2+\frac{\tau}{\lambda}\int_\Omega \sum_{i=1}^C w_i\|u-c_i\|_2^2+\\ &\quad \chi_{u\in[0,255]^3}+\chi_{Y(u)=I_g}\\ &=\text{argmin}_u \left\|u-\left(\frac{\tilde{u}+\frac{\tau}{\lambda}\sum_{i=1}^C w_i c_i}{1+\frac{\tau}{\lambda}}\right)\right\|_2^2+\\ &\quad \chi_{u\in[0,255]^3}+\chi_{Y(u)=I_g}\end{aligned} \quad (30)$$

It is equivalent to compute the projection of

$$\frac{\tilde{u}+\frac{\tau}{\lambda}\sum_{i=1}^C w_i c_i}{1+\frac{\tau}{\lambda}} \quad (31)$$

onto the intersection of the cube $[0,255]^3$ and the affine plane defined by $A.u = I_g$. There are many cases for this set: it can be a singleton, a triangle, a quadrilateral or a pentagon.

The computation of this projection is performed in two steps. The projection onto the plane is first achieved. Next the projection onto the intersection is computed. This two steps approach is justified by the theory, as shown by the following Lemma.

Lemma 1 (Two-steps projection.) *Let $\mathbf{x} \in \mathbb{R}^3$, and P an affine plane of $\mathbb{R}^3$, u the orthogonal projection of $\mathbf{x}$ onto P. Let $\mathcal{C}$ a non empty closed convex set included in P. And let v the orthogonal projection of u onto $\mathcal{C}$.*

So v is the orthogonal projection of $\mathbf{x}$ onto $\mathcal{C}$.

The first step (the projection onto the plane) has an algebraic expression given in the following Lemma.

Lemma 2 (Projection on the plane.) *Let $a \in \mathbb{R}^3$, $a \neq 0$, let $\alpha \in \mathbb{R}$, let*
$W := \{\mathbf{x} \in \mathbb{R}^n$ such that $\langle a|\mathbf{x}\rangle = \alpha\}$, and $s \in \mathbb{R}^n$.

So the orthogonal projection of s on the affine plane W is given by:

$$P_W(s) = \frac{a}{||a||_2}\left(\frac{\alpha - \langle a|s\rangle}{||a||_2}\right) + s. \tag{32}$$

Assume now that the projection onto the plane is computed and denoted by u'. There are two possibilities: u' is on the intersection or out of it. To determine this, the algorithm has to check if the point is inside the cube (by computing 6 inequalities). If u' is inside the cube, it is the final solution of the projection. Else, the final result is on an edge of the intersection. To determine the edge where it lies, the projection onto each edge is computed. The nearest projection among them is retained as final output.

Now, the complete algorithm computing the projection of a point onto the intersection of the cube $[0,255]^3$ and the affine plane defined by $A.u = I_g$ is given in Algorithm 3, assuming this intersection is a polygon defined by points $P_1, \ldots, P_n$. This algorithm has been described in [25].

Algorithm 3 Algorithm computing the proximal operator of $\chi_{[0,255]^3} + \chi_{\{u\in\mathbb{R}^3 \text{ st } Au=c\}}$.

$X \leftarrow \frac{A}{||A||^2}(I_g - \langle X|A\rangle) + X$ ▷ Orthogonal projection on the plane.
if $X \notin [0,255]^3$ **then**
  for $i = 1 : n-1$ **do**
    for $j = i+1 : n$ **do** ▷ For each edge.
      $\alpha \leftarrow \left\langle \overrightarrow{P_iP_j} | \overrightarrow{P_iX} \right\rangle / \left(\|\overrightarrow{P_iP_j}\|_2 \|\overrightarrow{P_iX}\|_2 \right)$
      if $\alpha > 1$ **then**
        $X_{i,j} \leftarrow P_j$
      else if $\alpha < 0$ **then**
        $X_{i,j} \leftarrow P_i$
      else
        $X_{i,j} \leftarrow P_i + \alpha\overrightarrow{P_iP_j}$
      end if
    end for
  end for
  $X \leftarrow \text{argmin}_{X_{i,j}} \|X - X_{i,j}\|_2$
end if

B About the Oblique Projection

In this section, the formulation of the oblique projection as the minimum of a convex functional is done. Next, a theoretical justification of the two steps in the oblique projection Algorithm 2 is provided.

Let $\mathcal{P}$ the vectorial plan defined by

$$\mathcal{P} := \{(r,g,b) \text{ such that } 0.299r + 0.587g + 0.114b = 0\}\ , \tag{33}$$

and $\mathcal{P}_l$ the affine plan defined by

$$\mathcal{P}_l := \{(r,g,b) \text{ such that } 0.299r + 0.587g + 0.114b = l\}\ , \tag{34}$$

Proposition 3 *The projection on the plan $\mathcal{P}_l$ of v in the direction of the vector*
$f_3 := \left(\frac{1}{3}, \frac{1}{3}, \frac{1}{3}\right)$ *is given by*

$$\text{argmin}_{u\in\mathcal{P}_l} \|A(u-v)\|_2^2\ , \tag{35}$$

with A a given matrix.

Proof Let $\{f_1, f_2\}$ an orthonormal spanning set of $\mathcal{P}$. The basis $\{f_1, f_2, f_3\}$ is denoted by f. It is a basis because $f_3 \notin \mathcal{P}$. $\{e_1, e_2, e_3\}$ denotes the canonical basis. With the Gram-Schmidt process the orthonormal basis $\left\{f_1, f_2, \tilde{f_3}\right\}$ denoted by $\tilde{f}$ is constructed. $\tilde{f_3} = \omega_1 f_1 + \omega_2 f_2 + \omega_3 f_3$ with $(\omega_1, \omega_2, \omega_3) \in \mathbb{R}^3$. Denote by P_e^f the matrix of the change of basis from e to f. Let us define the matrix A such that,

$$A = P_e^f \begin{pmatrix} 1 & 0 & \omega_1 \\ 0 & 1 & \omega_2 \\ 0 & 0 & \omega_3 \end{pmatrix} P_e^{f\,-1}. \tag{36}$$

It is easy to see that:

$$\forall v \in \mathcal{P}_l, \exists \lambda_1, \lambda_2 \in \mathbb{R} \text{ such that } v = \lambda_1 f_1 + \lambda_2 f_2 + 3l f_3. \tag{37}$$

Let us now compute:

$$\text{argmin}_{u\in\mathcal{P}_l} \|A(u-v)\|_2^2. \tag{38}$$

As $v \in \mathbb{R}^3$, $v = \lambda_1 f_1 + \lambda_2 f_2 + \lambda_3 f_3$.

For $u \in \mathbb{R}^3$, $u = \mu_1 f_1 + \mu_2 f_2 + \mu_3 f_3$. We write now:

$$\|A(u-v)\|_2^2 = \|(\lambda_1 - \mu_1)f_1 + (\lambda_2 - \mu_2)f_2 + \alpha(\lambda_3 - \mu_3)f_3\|_2^2. \tag{39}$$

$u \in \mathcal{P}_l \Rightarrow \mu_3 = 3l$, thus:

$$\begin{aligned} &\text{argmin}_{u\in\mathcal{P}_l} \|A(u-v)\|_2^2 \\ &\quad = \text{argmin}_{(\mu_1,\mu_2)\in\mathbb{R}^2} \|(\lambda_1 - \mu_1)f_1 + (\lambda_2 - \mu_2)f_2 + \\ &\qquad\qquad (\lambda_3 - 3l)(\omega_1 f_1 + \omega_2 f_2 + \omega_3 f_3)\|_2^2 \\ &\quad = \text{argmin}_{(\mu_1,\mu_2)\in\mathbb{R}^2} \|(\lambda_1 - \mu_1)f_1 + (\lambda_2 - \mu_2)f_2 + \\ &\qquad\qquad (\lambda_3 - 3l)\tilde{f_3}\|_2^2 \\ &\quad = \text{argmin}_{(\mu_1,\mu_2)\in\mathbb{R}^2} \|(\lambda_1 - \mu_1)f_1\|_2^2 + \|(\lambda_2 - \mu_2)f_2\|_2^2 \\ &\qquad + \|(\lambda_3 - 3l)\tilde{f_3}\|_2^2 \text{ because the basis is orthogonal.} \\ &\qquad\qquad = (\lambda_1, \lambda_2). \end{aligned} \tag{40}$$

Finally:

$$\text{argmin}_{u\in\mathcal{P}_l} \|A(u-v)\|_2^2 = \lambda_1 f_1 + \lambda_2 f_2 + 3l f_3\ , \tag{41}$$

which is the projection of v onto $\mathcal{P}_l$ in the direction of f_3.

Algorithm 4 Algorithm providing the solution of (42).

1: $u_1 \leftarrow P_{\mathcal{P}_l}(v)$
2: **if** $u_1 \in [0,255]^2$ **then**
3: $\quad u_0 \leftarrow u_1$
4: **else**
5: $\quad u_0 \leftarrow \text{argmin}_{u\in\mathcal{C}} \|u-u_1\|_2^2$
6: **end if**

In the following, we theoretically justify Algorithm 4.

Proposition 4 *Algorithm 4 provides the solution of* (42).

Proof We want now to compute:

$$u_0 := \text{argmin}_{u\in\mathcal{C}} \|A(u-v)\|_2^2. \tag{42}$$

First we write it as done in (40), which gives:

$$\begin{aligned} u_0 &= \text{argmin}_{u\in\mathcal{P}_l} \|A(u-v)\|_2^2 \\ &= \text{argmin}_{(\mu_1,\mu_2)\in\mathbb{R}^2} \|(\lambda_1-\mu_1)f_1\|_2^2 + \\ &\quad \|(\lambda_2-\mu_2)f_2\|_2^2 + \|(\lambda_3-3l)\tilde{f}_3\|_2^2 \ , \end{aligned} \tag{43}$$

because the basis is orthogonal.

The point u_0 is on the plan $\mathcal{P}_l$ because $\mathcal{C} \in \mathcal{P}_l$. First we compute $u_1 := \text{argmin}_{u\in\mathcal{P}_l} \|A(u-v)\|_2^2$. u_1 is on $\mathcal{P}_l \cap \{u \text{ such that } H(u) = H(v)\}$, which is a line. If $u_1 \in [0,255]^3$, $u_0 = u_1$ provides the result. If not, by remarking that

$$\begin{aligned} u_0 &= \text{argmin}_{u\in\mathcal{P}_l} \|A(u-v)\|_2^2 \\ &= \text{argmin}_{(\mu_1,\mu_2)\in\mathbb{R}^2} \|(\lambda_1-\mu_1)f_1\|_2^2 + \|(\lambda_2-\mu_2)f_2\|_2^2 \\ &\quad +\|(\lambda_3-3l)\tilde{f}_3\|_2^2 \\ &= \text{argmin}_{(\mu_1,\mu_2)\in\mathbb{R}^2} \|(\lambda_1-\mu_1)f_1\|_2^2 + \|(\lambda_2-\mu_2)f_2\|_2^2 \\ &\quad +\|(3l-3l)\tilde{f}_3\|_2^2 \\ &= \text{argmin}_{u\in\mathcal{C}} \|A(u-u_1)\|_2^2. \end{aligned} \tag{44}$$

thus u_0 is the closest point of $u_1 = P_{\mathcal{P}_l}(v)$ on $\mathcal{C}$.

i.e., Algorithm 4 computes the solution of (42).

References

1. Barnes, C., Shechtman, E., Finkelstein, A., Goldman, D.: Patchmatch: a randomized correspondence algorithm for structural image editing. ACM Transactions on Graphics **28**(3), 24 (2009)
2. Bay, H., Tuytelaars, T., Van Gool, L.: Surf: Speeded up robust features. In: European Conference on Computer Vision, pp. 404–417. Springer (2006)
3. Bredies, K.: Recovering piecwise smooth multichannel images by minimization of convex functionals with total generalized variation penalty. SFB Report **6** (2012)
4. Bresson, X., Chan, T.F.: Fast dual minimization of the vectorial total variation norm and applications to color image processing. Inverse Problems and Imaging **2**(4), 455–484 (2008)
5. Bugeau, A., Ta, V.T., Papadakis, N.: Variational exemplar-based image colorization. IEEE Transactions on Image Processing **23**(1), 298–307 (2014)
6. Chambolle, A.: An algorithm for total variation minimization and applications. Journal of Mathematical Imaging and Vision **20**(1-2), 89–97 (2004)
7. Chambolle, A., Pock, T.: A first-order primal-dual algorithm for convex problems with applications to imaging. Journal of Mathematical Imaging and Vision **40**(1), 120–145 (2011)
8. Charpiat, G., Hofmann, M., Schölkopf, B.: Automatic image colorization via multimodal predictions. In: D. Forsyth, P. Torr, A. Zisserman (eds.) European Conference on Computer Vision, pp. 126–139. Springer, Marseille, France (2008). URL `http://eccv2008.inrialpes.fr/`
9. Chen, T., Wang, Y., Schillings, V., Meinel, C.: Grayscale image matting and colorization. In: Asian Conference on Computer Vision, pp. 1164–1169 (2004)
10. Chen, Y., Ye, X.: Projection onto a simplex. arXiv preprint arXiv:1101.6081 (2011)
11. Combettes, P.L., Pesquet, J.C.: Proximal splitting methods in signal processing. In: Fixed-Point Algorithms for Inverse Problems in Science and Engineering, pp. 185–212. Springer (2011)
12. Combettes, P.L., Wajs, V.R.: Signal recovery by proximal forward-backward splitting. Multiscale Modeling & Simulation **4**(4), 1168–1200 (2005)
13. Di Blasi, G., Reforgiato, D.: Fast colorization of gray images. Eurographics Italian (2003)
14. Efros, A.A., Leung, T.K.: Texture synthesis by non-parametric sampling. In: IEEE International Conference on Computer Vision, vol. 2, pp. 1033–1038 (1999)
15. Goldluecke, B., Cremers, D.: An approach to vectorial total variation based on geometric measure theory. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 327–333 (2010)
16. Gonzales, R.C., Woods, R.E.: Digital image processing. Addison-Wesley Publishing Company (1993)
17. Gupta, R.K., Chia, A.Y.S., Rajan, D., Ng, E.S., Zhiyong, H.: Image colorization using similar images. In: ACM International Conference on Multimedia, pp. 369–378 (2012)
18. Hertzmann, A., Jacobs, C.E., Oliver, N., Curless, B., Salesin, D.H.: Image analogies. In: ACM Computer graphics and interactive techniques, pp. 327–340 (2001)
19. Horiuchi, T.: Colorization algorithm using probabilistic relaxation. Image and Vision Computing **22**(3), 197–202 (2004)
20. Irony, R., Cohen-Or, D., Lischinski, D.: Colorization by example. In: Eurographics conference on Rendering Techniques, pp. 201–210. Eurographics Association (2005)
21. Kolmogorov, A., Fomine, S.: Éléments de la théorie des fonctions et de l'anaylse fonctionelle (2nd edn.) mir (1977)
22. Levin, A., Lischinski, D., Weiss, Y.: Colorization using optimization. ACM Transactions on Graphics **23**(3), 689–694 (2004)
23. Levinshtein, A., Stere, A., Kutulakos, K.N., Fleet, D.J., Dickinson, S.J., Siddiqi, K.: Turbopixels: Fast superpixels using geometric flows. IEEE Transactions on Pattern Analysis and Machine Intelligence **31**(12), 2290–2297 (2009)
24. Nikolova, M., Steidl, G.: Fast hue and range preserving histogram specification: Theory and new algorithms for color image enhancement. Preprint (2013)
25. Pierre, F., Aujol, J.F., Bugeau, A., Ta, V.T., Papadakis, N.: Exemplar-based colorization in rgb color space. IEEE International Conference on Image Processing (2014)

26. Quang, M.H., Kang, S.H., Le, T.M.: Image and video colorization using vector-valued reproducing kernel hilbert spaces. Journal of Mathematical Imaging and Vision **37**(1), 49–65 (2010)
27. Raguet, H., Fadili, J., Peyré, G.: Generalized forward-backward splitting. arXiv preprint arXiv:1108.4404 (2011)
28. Rudin, L.I., Osher, S., Fatemi, E.: Nonlinear total variation based noise removal algorithms. Physica D: Nonlinear Phenomena **60**(1), 259–268 (1992)
29. Sapiro, G.: Inpainting the colors. In: IEEE International Conference on Image Processing, vol. 2, pp. II–698 (2005)
30. Takahama, T., Horiuchi, T., Kotera, H.: Improvement on colorization accuracy by partitioning algorithm in cielab color space. In: Advances in Multimedia Information Processing-PCM 2004, pp. 794–801. Springer (2005)
31. Wei, L.Y., Levoy, M.: Fast texture synthesis using tree-structured vector quantization. In: ACM Computer graphics and interactive techniques, pp. 479–488. Press/Addison-Wesley Publishing Co. (2000)
32. Welsh, T., Ashikhmin, M., Mueller, K.: Transferring color to greyscale images. ACM Transactions on Graphics **21**(3), 277–280 (2002)
33. Yatziv, L., Sapiro, G.: Fast image and video colorization using chrominance blending. IEEE Transactions on Image Processing **15**(5), 1120–1129 (2006)
34. Yin, S., Cao, L., Ling, Y., Jin, G.: One color contrast enhanced infrared and visible image fusion method. Infrared Physics & Technology **53**(2), 146–150 (2010)