



HAL
open science

Luminance-Hue Specification in the RGB Space

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

► **To cite this version:**

Fabien Pierre, Jean-François Aujol, Aurélie Bugeau, Vinh-Thong Ta. Luminance-Hue Specification in the RGB Space. Scale-Space and Variational Method in Computer Vision, Jun 2015, Lège Cap-Ferret., France. 10.1007/978-3-319-18461-6_33 . hal-01168528

HAL Id: hal-01168528

<https://hal.science/hal-01168528>

Submitted on 26 Jun 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Luminance-Hue Specification in the RGB Space

Fabien Pierre^{1,2} *, Jean-François Aujol¹, Aurélie Bugeau², and Vinh-Thong Ta³

¹Univ. Bordeaux, IMB, CNRS, UMR 5251, F-33400 Talence, France.

²Univ. Bordeaux, LaBRI, CNRS, UMR 5800, PICTURA, F-33400 Talence, France.

³Bordeaux INP, LaBRI, CNRS, UMR 5800, PICTURA, F-33400 Talence, France.

Abstract. This paper is concerned with a problem arising when editing color images, namely the Luminance-Hue Specification. This problem often occurs when converting an edited image in a given color-space to RGB. Indeed, the colors often get out of the standard range of the RGB space which is commonly used by most of display hardwares. Simple truncations lead to inconsistency in the hue and luminance of the edited image. We formalize and describe this problem from a geometrical point of view. A fast algorithm to solve the considered problem is given. We next focus on its application to image colorization in the RGB color-space while most of the methods use other ones. Using directly the three RGB channels, our model avoids artifact effects which appear with other color-spaces. Finally a variational model that regularizes color images while dealing with Luminance-Hue Specification problem is proposed.

Keywords: Color, Luminance-Hue Specification, Color-spaces, Colorization, Total Variation, Optimization

1 Introduction

Color images can be characterized by their luminance, hue and saturation. The hue represents the human perception of the pure color while the saturation represents how the color is mixed to the white color. Many editing operations can be performed on the colors. For instance one may want to modify the hue while keeping the luminance, or conversely, to modify the luminance while keeping the hue constant. These operations are not as trivial as they seem. They can be performed in a hue-saturation-value color-space (*e.g.*, HSV) or luminance-chrominance space (*e.g.*, YUV) in which it is easy to modify or specify one channel while keeping the two others constant. Unfortunately the conversion back to RGB does not ensure the computed values of the three channels H, S, or Y are maintained. In most of imaging problems, the final RGB color has to be maintained between 0 and 255 in order to be displayed onto the screen of a device.

* 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.









Initial color	Initial Y	Initial hue	Specified Y	Final after truncation	Final Y	Final hue	Method of [?]	Hue of [?]	Our result
	90	28	40		47	10		6	
	180	84	40		58	120		62	

Table 1. The original color is first transformed into the YUV color-space. The value of Y is then specified to 40 and the YUV values are turned back to RGB with standard linear transformation. When this transformation is out of range, the final results is obtained with truncations which leads to some inconsistent results.

If the conversion of the HSY image does not remain in the RGB standard range, the channels are truncated in order to stay in this range. This truncation is done without paying attention neither to the hue or the luminance. Obviously this problem arises with any color spaces and in particular to luminance-chrominance spaces (*i.e.*, Lab, YUV or $\alpha\beta$).

Table ?? proposes an example to observe the problem in the YUV color-space. We illustrate the specification of the same luminance for two different colors. Both colors are first converted into the YUV color-space. In this space, the channel Y is changed to the specified value. Finally, we transform back the color into the RGB color-space. Theoretically, the luminance, chrominances and even the hue should not be changed after this conversion. Nevertheless, the range of the RGB color-space was not respected. The final result is obtained by truncation of coordinates, leading to a change of the luminance and the hue. The change of the hue during conversions between color-spaces highlights an important and common problem. In particular, in the real world, a change of lighting of an object should not change its hue. The method of [?] performs an orthogonal projection onto the constraint of constant luminance, that provides a color with the required luminance. Nevertheless, the hue is not preserved. The method proposed in this paper tackles with this issue which can be summarized as follows.

Problem. Given an RGB color, can we compute another color with the same hue, a specified luminance and which lives in the standard range of the RGB color-space ? The answer to this question will turn out to be yes as shown in Section ???. We will refer to this problem as the *Luminance-Hue Specification* in the rest of the paper.

This problem has already been addressed for image enhancement in [?, ?, ?, ?]. The objective is to specify the histogram of the intensity channel of a given image in order to enhance it. After specifying the intensity histogram, the final color image is computed by taking care of keeping the original hue. The algorithm proposed in [?] will be detailed in section ??? and compared to ours. Contrary to this previous approach, we will give a geometric point of view of the Luminance-Hue Specification problem and propose a fast algorithm to solve it.

In this paper, we also focus the application of the Luminance-Hue Specification to image colorization. Image colorization consists in turning a gray-scale image into a color one by adding some color information to each pixel. 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 and equals to the original one.

To add color information, 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 [?, ?, ?]. The second category, called exemplar-based methods, 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*). [?, ?, ?] propose exemplar-based methods with regularized results.

To colorize images, most approaches keep the luminance constant and compute the chrominance channels of a luminance-chrominance space (*e.g.*, Lab, YUV or $\alpha\beta$). As explained previously, the final transformation from this luminance-chrominance space into the RGB does not guarantee the consistency of the solution. Changes of the luminance or the hue may occur. A solution to avoid any range problems in the final result is to work directly in the RGB color-space. To our best knowledge, only three colorization methods work in the RGB color-space: [?], [?] and [?]. For instance [?] limits the choice of color for a given luminance to all colors having this luminance. This method is therefore very difficult to adapt to the exemplar-based colorization where the color extracted from the source image may have a different luminance. [?] proposes a variational model working in the RGB color-space. This method does not ensure the preservation of the luminance, leading to blur effects.

In this paper we propose to solve the Luminance-Hue Specification problem within the seminal work of [?]. More specifically we propose an exemplar-based colorization method which works directly in the RGB color-space. The best RGB vector for each pixel of the target image is chosen within the source image. We further apply our new Luminance-Hue Specification algorithm to obtain a coherent color. Since the proposed approach works in the RGB color-space, we do not convert the images into a luminance-chrominance space and the result naturally stays in the standard range to be displayed.

To go further with image colorization, this paper also presents in detail a new model for color image regularization. It is based on the Rudin Osher and Fatemi functional [?] with a luminance constraint. The final algorithm is inspired by a primal-dual scheme [?]. The proposed regularization method is able to couple the R, G and B channels which encourages clean contours in the final result. The control of the hue during the optimization process enables realistic regularized images.

Contributions and Outline. First, the Luminance-Hue Specification problem is formalized and described from a geometrical point of view. Next, a fast algorithm inspired from the Smit’s algorithm for ray-tracing [?] is proposed to solve this problem. Its application to image colorization is later presented with a

non-regularized exemplar-based colorization method. Color image regularization with a luminance constraint is finally addressed by designing an optimization scheme. To that end, the Luminance-Hue Specification algorithm is integrated into an iterative primal-dual algorithm.

2 The Luminance-Hue Specification Problem

In this section, we present the Luminance-Hue Specification problem from a geometrical point of view and we propose an efficient algorithm to solve it.

2.1 The HSI Color-space

The RGB color-space is not adapted to the human perception and description of colors. To tackle with this issue, particular color-spaces have been created. For instance, the HSV color-space defines colors with three particular channels: 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 [?]) for its simplicity in terms of geometric interpretation. This point of view has been used recently for image enhancement by Nikolova *et al.* [?].

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

$$H = \begin{cases} \theta & \text{if } B \leq G \\ 360 - \theta & \text{otherwise,} \end{cases} \quad (1)$$

where θ is an angle in degrees, such as:

$$\theta = \arccos \left\{ \frac{\frac{1}{2}((R - G) + (R - B))}{\sqrt{(R - G)^2 + (R - B)(G - B)}} \right\}. \quad (2)$$

The saturation S represents how the color is mixed with the white color and is defined as:

$$S = 1 - \frac{3}{R + G + B} (\min(R, G, B)). \quad (3)$$

It is equal to 0 if the color is gray, and has its maximum value if the color is away from any gray color.

The value of the intensity I can be compared to the luminosity 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:

$$I = \frac{R + G + B}{3}. \quad (4)$$

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

Another way to model the luminosity of a scene is the luminance defined by:

$$Y = 0.299R + 0.587G + 0.114B. \quad (5)$$

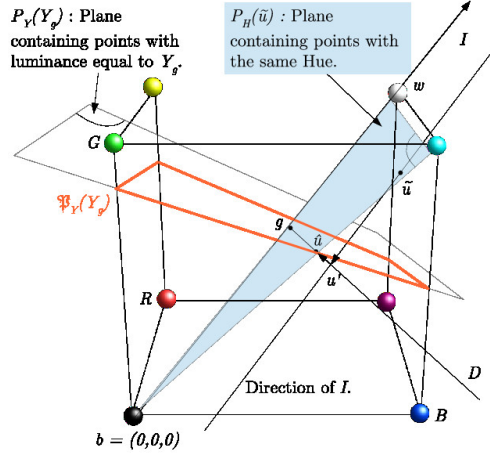


Fig. 1. The projection of \tilde{u} on the convex set $\mathfrak{P}_Y(Y_g)$ is computed in two steps. First the oblique projection of \tilde{u} on D in the direction of I (i.e., the diagonal of the cube) is computed and denoted by u' . Secondly, the algorithm computes \hat{u} , the closest point of u' onto the convex set $D \cap [0, 255]^3$.

2.2 An Oblique Projection from a Geometrical Point of View

In this section an oblique projection \tilde{P}_G , maintaining the hue constant and specifying a luminance, is formalized. Fig. ?? shows elements described below.

Let us first introduce some notations and properties.

Proposition 1 *All the points in the RGB color-space which have the same hue as a given color \tilde{u} live on the same open half-plane denoted by*

$$P_H(\tilde{u}) := \{u' \in \mathbb{R}^3 \text{ such that } H(\tilde{u}) = H(u')\}. \quad (6)$$

Proof. See, e.g., [?]. □

The half-plane $P_H(\tilde{u})$ (blue in Fig. ??) contains \tilde{u} and is bounded by the line I passing through coordinates $(0, 0, 0)$ (black b) and $(255, 255, 255)$ (white w).

In order to specify a luminance, let us define the set of all colors for which the luminance is equal to a given scalar Y_g . According to equation (??), this set is a plane denoted by:

$$P_Y(Y_g) := \{u \in \mathbb{R}^3 \text{ such that } Y(u) = Y_g\}. \quad (7)$$

Since the final result has to be displayed, it has to be in the standard range of the RGB space. Let us define the set of possible specified results:

$$\mathfrak{P}_Y(Y_g) := \{u \in [0, 255]^3 \text{ such that } Y(u) = Y_g\}. \quad (8)$$

To solve the Luminance-Hue Specification problem, we define an oblique projection. Its result has the same hue as the original color (denoted by \tilde{u}) and its

Algorithm 1 Algorithm for oblique projection, inspired by ray-tracing.

```

1:  $(R, G, B)$  is the input.  $Y_g$  is the target luminance.  $(\hat{R}, \hat{G}, \hat{B})$  is the output.
2:  $Y \leftarrow 0.299R + 0.587G + 0.114B$ 
3:  $(\hat{R}, \hat{G}, \hat{B}) \leftarrow (R, G, B) - (Y, Y, Y) + (Y_g, Y_g, Y_g)$ 
4: if  $(\hat{R}, \hat{G}, \hat{B}) \in [0, 255]^3$  then
5:    $(\hat{R}, \hat{G}, \hat{B})$  is the result.
6: else
7:    $T \leftarrow \emptyset$ 
8:   for  $K \in \{\hat{R}, \hat{G}, \hat{B}\}$  do
9:     if  $K > 255$  then
10:       $T \leftarrow T \cup \left\{ \frac{255 - Y_g}{K - Y_g} \right\}$ 
11:     end if
12:     if  $K < 0$  then
13:       $T \leftarrow T \cup \left\{ \frac{-Y_g}{K - Y_g} \right\}$ 
14:     end if
15:   end for
16:    $t_{min} \leftarrow \min T$ 
17:    $(\hat{R}, \hat{G}, \hat{B}) \leftarrow (1 - t_{min}) \cdot (Y_g, Y_g, Y_g) + t_{min} \cdot (\hat{R}, \hat{G}, \hat{B})$ 
18: end if

```

luminance is equal to Y_g . Thus, the result belongs to the half-line $P_Y(Y_g) \cap P_H(\tilde{u})$ denoted by D . An oblique projection onto $P_Y(Y_g)$ in the direction of I can solve the problem. To compute it, we define γ such that $Y[\tilde{u} + \gamma(1, 1, 1)] = Y_g$, leading to $\gamma = Y_g - Y(\tilde{u})$, and the first result $u' = \tilde{u} + \gamma(1, 1, 1)$. After this first step there are two possibilities: u' is inside the cube $[0, 255]^3$ or it is not. If it is in the cube, no further processing is needed. Otherwise the Smit's algorithm for ray-tracing [?] is used to compute the contact point of the RGB cube $[0, 255]^3$ and D (See Algorithm ?? for details). This algorithm uses a parametrization of D for computing its intersection with the cube.

2.3 Equivalence with Nikolova *et al* [?]

In this section the oblique projection is compared to the additive algorithm of Nikolova *et al* [?] used for image enhancement. The problem is to compute a color with a given hue and intensity, and living in the standard range of the RGB color-space. The only difference with ours is that we specify luminance and not intensity. This algorithm is directly adaptable for our problem by replacing the constraint of intensity by the constraint of luminance.

Proposition 2 *Algorithm ?? and additive algorithm of [?] provide the same result.*

Proof. Let us denote the input (R, G, B) . In the case when $(R, G, B) - (Y, Y, Y) + (Y_g, Y_g, Y_g)$ is in $[0, 255]^3$ it is straightforward, because the algebraic expression of the result is exactly the same for the two algorithms.

In the case when this point is out of the cube, the two algorithms process a point in one face of the cube. Remark that this point is only obtained by multiplying the vector (R, G, B) by a scalar and adding an equal value to each component. Next, the resulting point is onto the half-plane $\{u \in \mathbb{R}^3 \text{ such that } H(u) = H(R, G, B)\}$. Moreover, the luminance of the final result is equal to Y_g in the two cases. Thus, the result lives on a half-line D that is the intersection of these two planes. D is not contained in one of the faces of the cube. Moreover, one of the coordinates of the result is equal to 0 or 255 for the two algorithms and the others are in the cube. Thus, the final result is at the intersection of a face of the cube $[0, 255]^3$ and the half-line D . This last intersection is unique, thus, the results are the same for the two algorithms. Let us remark that the two algorithms perform almost the same computations. In its formulation, Algorithm ?? is closer of the geometric interpretation than the additive algorithm of [?]. \square

2.4 Application to Non-Regularized Colorization

Welsh *et al.* [?] propose a patch-based method to colorize images. This method works in the $\alpha\beta$ color-space. This method has been extended by Bugeau *et al.* [?] in the YUV color-space. In these two methods, the final result of the colorization process may not be in the standard range of RGB. By working directly in the RGB color-space, we avoid this problem. To colorize an image, we first compute a mapping between the target image and the gray-scale version of the source. We then extract the RGB vectors for each pixel and finally specify the luminance value with Algorithm ??.

Fig. ?? shows two colorization examples, each from a source image and a target one to be colorized. To obtain the results, we consider for each pixel of the target image, the patch around this pixel. We then search in the luminance channel of the source image the closest patch using the PatchMatch algorithm [?]. The final result is the RGB color of the source image at this closest patch central pixel. The comparison of patches is performed with the L^1 norm between standard-deviations (see, *e.g.*, distance f_1 used in [?]). This result leads to blurring effects since it does not respect the Y channel. The specified image provides a more consistent result (column (d)). The specified luminance is directly the target image.

3 RGB Variational Model for Image Colorization

In order to improve colorization results, we now propose a variational model and an optimization scheme, holding the luminance specification constraint.

3.1 A Variational Approach

Assume that c is an image, totally or partially colored. We want to regularize or to inpaint the color image by minimization of the Total Variation (TV). Moreover, we want to constrain the final solution to have a specified luminance.

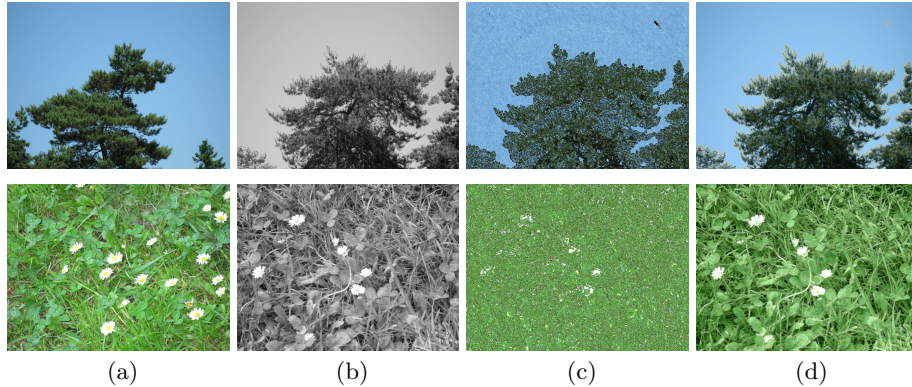


Fig. 2. (a) Source images. (b) Target images. (c) Exemplar-based colorization in the RGB space. (d) represents Image (c) with specified Y (Algorithm ??). Exemplar-based colorization by simple mapping with and without luminance specification. The Y grayscale image contains essential information.

A standard approach for this problem is the use of a variational model. We introduce a functional where $u = (R, G, B)$ stands for a RGB image:

$$\inf_u TV_{RGB}(u) + \lambda/2 \|M(u - c)\|_2^2 + \chi_{u \in [0,255]^3} + \chi_{Y(u)=Y_g}, \quad (9)$$

where

$$TV_{RGB}(u) = \int_{\Omega} \sqrt{\gamma \partial_x Y_g(\mathbf{x})^2 + \gamma \partial_y Y_g(\mathbf{x})^2 + \sum_{K=R,G,B} \partial_x K(\mathbf{x})^2 + \partial_y K(\mathbf{x})^2} dx. \quad (10)$$

The term $\|M(u - c)\|_2^2$ is the data fidelity term. M is a mask whose value is equal to 0 if no color is defined at the current pixel, 1 otherwise. With this formulation, the data term may be not complete. TV_{RGB} is the regularization term. The parameter γ enforces the coupling of the Y channel with the RGB ones in order to preserve contours (see, *e.g.*, [?] for the influence of γ). Y_g is the luminance to be specified. The parameter λ controls the trade-off between the regularization and the data fidelity term. The χ functions model the constraints. Its values are 0 if the constraint is respected and $+\infty$ otherwise. The constraint $Y(u) = Y_g$ means that the luminance of the final result is equal to the specification Y_g . This constraint preserves the initial image content. $\chi_{u \in [0,255]^3}$ ensures that the final result is in the standard range of RGB space.

3.2 Minimization of the RGB Model

To minimize problems related to the Total Variation, a primal-dual formulation is proposed and solved by Chambolle and Pock [?]. The FISTA algorithm of Beck *et al.* [?] could also minimize this model. Our experiment has shown a

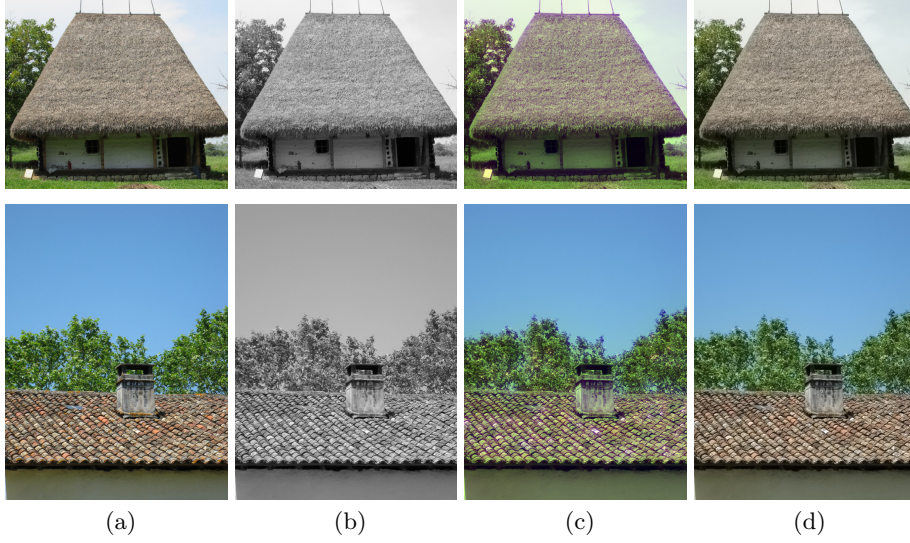


Fig. 3. (a) Original color image. (b) Luminance of (a) computed by (?). (c) Method of [?], equivalent to Algorithm ??. (d) Result obtained with Algorithm ??. The minimization of the model (??) (Algorithm ??) leads to some unrealistic colors different from the original ones. The control of the hue during the iterative process performs relevant results (Algorithm ??).

faster convergence of the primal-dual algorithm for this problem. Applied to our model, the primal-dual algorithm reads as Algorithm ??. P_B represents the projection onto L^2 unit ball due to the dual formulation of TV [?]. P_G is the orthogonal projection onto $\mathfrak{P}_Y(Y_g)$ described in [?]. The discrete gradient ∇ and divergence div operators are defined as in [?]. Time-steps are fixed to $\sigma = 10^{-3}$ and $\tau = 20$.

Since the functional is convex, our algorithm computes a solution of the problem (?). At this point, the algorithm does not produce satisfying results. This problem is highlighted in Fig. ?? where we propose to regularize the data image but holding its luminance channel constant. We choose $\gamma = 25$, $\lambda = 1$ and $M = Id$, the identity operator. Algorithm ?? is applied with the orthogonal pro-

Algorithm 2 Primal-dual algorithm minimizing the RGB model (?).

- 1: $p_0 \leftarrow 0$
 - 2: **for** $n \geq 0$ **do**
 - 3: $p_{n+1} \leftarrow P_B(p_n + \sigma \nabla \bar{u}_n)$
 - 4: $u_{n+1} \leftarrow P_G \left(M \frac{u_n + \tau (\text{div}(p_{n+1}) + \lambda c)}{1 + \tau \lambda} + (1 - M)(u_n + \tau \text{div}(p_{n+1})) \right)$
 - 5: $\bar{u}_{n+1} \leftarrow 2u_{n+1} - u_n$
 - 6: **end for**
-

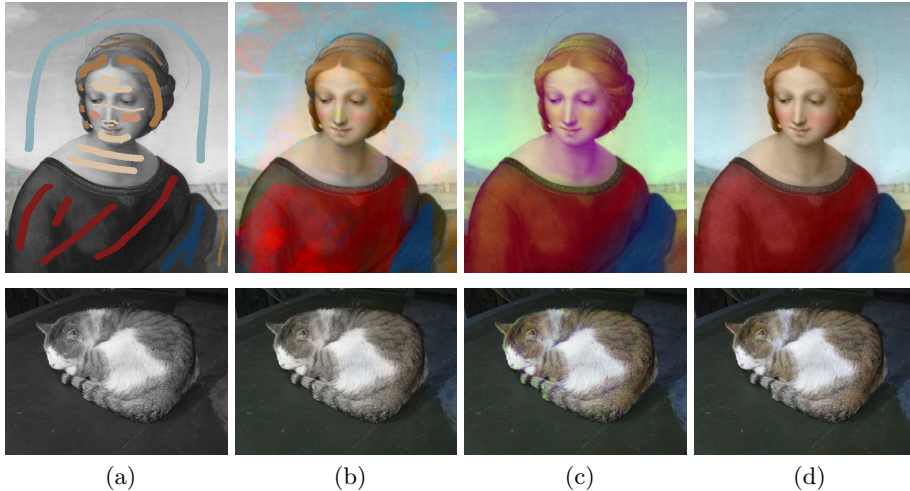


Fig. 4. (a) Scribbles. (b) TV on chrominances. (c) Result obtained with Algorithm ?? . (d) Result obtained with Algorithm ?? . Manual colorization with diffusion with TV. Neither the TV on the chrominance channels, nor the classical TV-L2 model (??) are able to perform realistic results. With a control of the hue during the iterative process, Algorithm ?? provides a relevant solution. Remark that scribbles on the cat are not easy to see because they are one pixel sized and sampled in a regular grid on 1% of the image.

jection (Fig. ??). The contours are well respected and the images are not drab, but they are now too shiny, containing irrelevant colors (green or violet) due to the orthogonal projection P_G that does not preserve the hue. To tackle this issue, we replace the projection P_G in Algorithm ?? by \tilde{P}_G , the Luminance-Hue Specification Algorithm ??, leading to Algorithm ?? . With this new algorithm, the iterative process produces realistic images with natural colors. For example, the textures do not become green (see Fig. ??). This figure is an artificial experiment to illustrate the interest of Algorithm ?? compared to [?]. We also propose, in Fig. ?? to inpaint the colors of an image. Some colors are given by the user, then the algorithm diffuses them to recover a complete image. In this case the value of the mask M is equal to 1 where the user puts a color and 0

Algorithm 3 Primal-dual algorithm inspired from Algorithm ??.

- 1: $p_0 \leftarrow 0$
 - 2: **for** $n \geq 0$ **do**
 - 3: $p_{n+1} \leftarrow P_B(p_n + \sigma \nabla \bar{u}_n)$
 - 4: $u_{n+1} \leftarrow \tilde{P}_G \left(M \frac{u_n + \tau (\text{div}(p_{n+1}) + \lambda c)}{1 + \tau \lambda} + (1 - M)(u_n + \tau \text{div}(p_{n+1})) \right)$
 - 5: $\bar{u}_{n+1} \leftarrow 2u_{n+1} - u_n$
 - 6: **end for**
-

Method	TV on chrominances	Result of Algorithm ??	Result of Algorithm ??
PSNR	28.9	31.8	37.4

Table 2. Comparison of PSNR for techniques presented in Fig. ?? for the image of the cat. This quantitative comparison highlights the quality of the results provided by Algorithm ??

otherwise. From left to right, one can see the original image with scribbles, the result obtained with an inpainting of chrominance channels, the minimization of model (??) (Algorithm ??) and the result of Algorithm ?. In the first case, the chrominance channels are diffused with the Total Variation without taking care of the initial luminance contours. Moreover the result is more drab for the cat. We choose $\gamma = 35$ and $\lambda = 10^{-5}$. In the case of the minimization of (??), we can see that the orthogonal projection does not respect the original palette to colorize the image. For example, the edge of the clothe of the *Madonna* is green, although there is no green color among scribbles at this location. Moreover, her face is violet, which does not correspond to any scribble, and it appears unnatural. By replacing the orthogonal projection by the Luminance-Hue Specification Algorithm ?, we solve the problem. The final result has respected contours and realistic colors. In the case of the cat, the original image is available. The scribbles are sub-sampled from it in a sparse regular grid. We compare the PSNR of the results obtained with the three different techniques. The values are summarized in Table ?. This measure confirms numerically the superiority of Algorithm ?.

Conclusion

In this paper, we propose an oblique projection, able to solve the Luminance-Hue Specification problem. We also show that a classical variational model fails for the regularization of images with the luminance specification. We finally see that the control of the hue during the iterative process for the minimization of the first variational model performs suitable results. This algorithm could be extended to correction of chromatic artefacts in the problem of dehazing or to the reduction of JPEG color artefacts.