Recent Approaches for Image Colorization

Fabien Pierre, Jean-François Aujol

Recent Approaches for Image Colorization

Fabien Pierre and Jean-François Aujol

Abstract In the last years, image and video colorization has been considered from many points of view. The technique consists in the addition of a color component to a gray-scale image. This operation needs additional priors which can be given by manual intervention of the user from an example image or be extracted from a large dataset of color images. A very large variety of approaches has been used to solve this problem, like PDE models, non-local methods, variational frameworks, learning approaches, etc. In this chapter, we aim at providing a general overview of state-of-the-art approaches with a focus on few representative methods. Moreover, some recent techniques from the different types of priors (manual, exemplar-based, dataset-based) are explained and compared. The organization of the chapter aims at describing the evolution of the techniques in relation to each other. A focus on some efficient strategies is proposed for each kind of methodology.

1 Context and Modeling

1.1 Challenge

Image colorization consists in the transformation of a grayscale image into a color one. The reverse transformation, *i.e.*, turning a color image into a grayscale one is based on visual assumptions and it is also an active research topic [43, 18, 74]. Image colorization is useful for the entertainment industry to make old film productions attractive to young people, for instance. In France, in 2014, *Apocalypse*, a historical

Fabien Pierre
LORIA, UMR CNRS 7503, Université de Lorraine, INRIA projet Magrit, France,
e-mail: fabien.pierre@loria.fr

Jean-François Aujol
Univ. Bordeaux, Bordeaux INP, CNRS, IMB, UMR 5251, F-33400 Talence, France,
e-mail: jean-francois.aujol@math.u-bordeaux.fr

documentary by I. Clarke and D. Costelle was made from archives colorized by F. Montpellier of the ImaginColor company. The broadcast gathered over 18.5% of viewers over the age bracket 11-14 during the first two episodes [46]. The colorization for movies is mostly performed manually, which is a very tedious work. As an example, the colorization of about four hours of video sequences for the *Apocalypse* documentary required forty-seven weeks by F. Montpellier and his team. Image colorization can also be used to help a user to analyze an image, for example for sensor fusion in [85]. For instance, to assist in airport security screening, color is added to the X-ray scanner result based on the density of the objects, so that the operator can know their composition and quickly interpret the result [1]. Image colorization can also be used to restore artistic heritage, for example [27] or [81]. This old subject started with the ability of screens to display color. A first approach, very basic, consists in matching each grayscale to a color [28]. However, it is impossible to recover every color without additional information (there are 256 grey levels and about 16 million colors displayable on standard screens). In existing approaches, this information can be added by three ways: the first one directly adds color to the image by the user (see for example the approach of [48]), the second one provides an example image (also called source image, see for example the method of [79]), and the third one uses a deep learning approach based on a large database (see for instance the method of [83]).

In this chapter, we propose a general overview of colorization methods which have been described in the literature with a focus on few representative approaches. This review is not based on the application point of view but it has been done from a methodological perspective. The term "automatic" has been widely used, but it means in fact that the algorithms are able to assist the user. For manual methods, the diffusion of the colors put by the user is automatic, for exemplar-based approaches, the diffusion of colors from a given reference image to target one is automatic but actually it requires the choice of the source image. For dataset-based colorization, the colorization is automatic after training on a large dataset given by the user. In this chapter, an overview of the three different approaches to colorize images (manual, exemplar-based and dataset-based) is proposed. In particular, a highlight on a variational model is used as a thread along the chapter because this model enables some coupling of different approaches such as manual with exemplar-based. More generally, we focus on different strategies available among state-of-the-art methods for each kind of methodology. Moreover, a final section proposes an overview of coupled strategies.

In this chapter, the mathematical modeling of the colorization problem is reviewed in Section 1.2. Next, in Section 1.3, we recall the definition of the range of the solution and we present an algorithm to compute an orthogonal projection onto this set. The three next Sections deal with, respectively, the manual, the exemplar-based and the dataset-based colorization. Finally, in Section 5, we propose an overview about the coupling of some techniques within a variational formulation.

1.2 Mathematical Modeling of Colorization

In order to model the colorization problem, let us consider the luminance-chrominance color spaces. The results of this section are based on the papers [64, 68] that can be considered as state-of-the-art for luminance specification. In all state-of-the-art approaches, the gray scale image is considered as the luminance channel of a color image. The luminance can be defined as a weighted average of the RGB channels:

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

Some other definitions are also sometimes used. For instance, the L channel of the CIE Lab color space can be used. In order to preserve its content, colorization methods must always require that the luminance channel of the image of interest is equal to the target image. Most methods compute only the two chrominance channels, complementary to the luminance, which is enough to provide a displayable color image.

Some different spaces have been introduced, such as YUV, YCbCr, YIQ etc. The transformation from RGB to YUV is linear and defined with the following matrices:

$$R, G, B, Y \in [0, 255], U \in [-111.18, 111.18], V \in [-156.825, 156.825]. \qquad (2)$$

$$\begin{pmatrix} Y \\ U \\ V \end{pmatrix} = \begin{pmatrix} 0,299 & 0,587 & 0,114 \\ -0,14713 & -0,28886 & 0,436 \\ 0,615 & -0,51498 & -0,10001 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix}. \qquad (3)$$

Let us notice that the main problem raised by these color spaces is that all the luminance-chrominance values cannot be converted into a RGB color between 0 and 255. Thus, some additional techniques have to be employed to recover the RGB color image [64]. These techniques are out of the scope of this chapter, but the reader has to keep in mind that they are essential to compute the final result. The next section recall the basis of gamut problem in the case of the YUV color space.

1.3 Range of Chrominance

The natural problem arising when editing a color while keeping its luminance or intensity constant, is the preservation of the RGB standard range of the produced image. Most of the methods of the literature work directly in the RGB space [55, 26, 64], since it is easier to maintain the standard range. Nevertheless, working in the RGB space needs to process 3 channels, while 2 chrominance channels are enough to edit a color image while keeping the luminance.

1.3.1 Description of the Range

In this section, we recall the geometric description of the set of chrominance values which correspond to a particular luminance level, and which are contained in the RGB standard range. Let us denote by $T(y, u, v)$ the invertible linear operator mapping YUV colors onto RGB ones. This operator corresponds to the inverse of the operation described in Equation (3).

Proposition 1 *Let y be a value of luminance between 0 and 255. The set of chrominance values (u, v) that satisfy $T(y, u, v) \in [0, 255]^3$ is a convex polygon.*

Remark 1 : For a given luminance, the chrominance values out of this polygon can be transformed into the RGB space, but they are out of the bounds of the RGB cube. A truncation of the coordinates is usually done, but it generally changes both the luminance and the hue of the result.
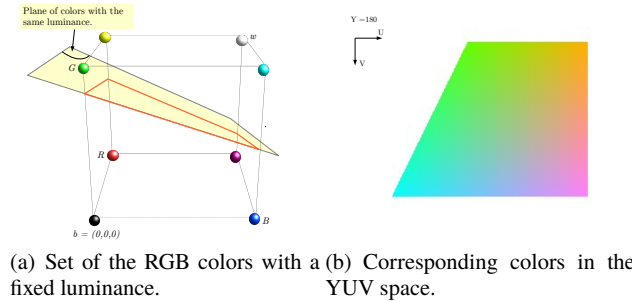

(a) Set of the RGB colors with a fixed luminance. (b) Corresponding colors in the YUV space.

Fig. 1 The set of the RGB colors with a particular luminance is a convex polygon. The map from RGB to YUV being affine, the set of the corresponding chrominances is also a convex polygon.

Proof (of Proposition 1) The intuition of the proof is given in Figure 1. The set of the colors in the RGB cube whose luminance is equal to a particular value y is a convex polygon (see, *e.g.*, [64]). Indeed, the set of colors with a particular luminance is an affine plane in $\mathbb{R}^3$ and the intersection of the RGB cube with it is a polygon. The transformation of the RGB values into the YUV space being affine, the set of corresponding colors is thus also a convex polygon included in the set $Y = y$.   □

1.3.2 Orthogonal Projection Onto the Convex Range

Pixel-wise, the valid chrominances are contained in a convex polygon that has, at most, 6 edges. The numerical computation of the vertex coordinates has been detailed in [68]. When the vertices are computed, and denoted by P1, P2, *etc*, the orthogonal projection onto the polygon is computed as follows.

The algorithm first checks if the corresponding RGB value is between 0 and 255. If so, the point is its own orthogonal projection. If not, the orthogonal projection is onto one of the edges and can be computed for each of them. Finally, the closest result is retained as the solution. The algorithm is summarized in Algorithm 1 and illustrated in Figure 2.

Algorithm 1 Algorithm computing projection $P_{\mathcal{R}}$.

Require: X: chrominance vector; Y luminance value.
1: **if** $RGB(Y, X) \notin [0, 255]^3$ **then**
2:     **for** $i = 1 : n$ **do**
3:         $j \leftarrow i + 1 \bmod n$
4:         $\alpha \leftarrow \left\langle \overrightarrow{P_i P_j} \mid \overrightarrow{P_i X} \right\rangle / \left(\| \overrightarrow{P_i P_j} \|_2 \right)$
5:         **if** $\alpha > 1$ **then**
6:             $X_{i,j} \leftarrow P_j$
7:         **else if** $\alpha < 0$ **then**
8:             $X_{i,j} \leftarrow P_i$
9:         **else**
10:            $X_{i,j} \leftarrow P_i + \alpha \overrightarrow{P_i P_j}$
11:         **end if**
12:     **end for**
13:     $X \leftarrow \mathrm{argmin}_{X_{i,j}} \| X - X_{i,j} \|_2$
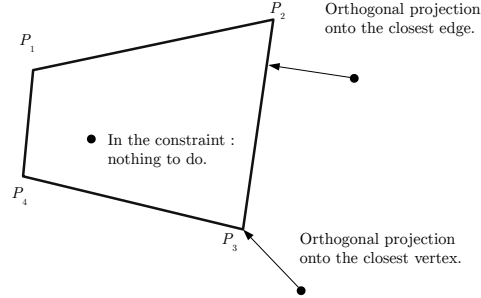14: **end if**

Fig. 2 To compute the orthogonal projection, different cases can appear. If the YUV color respects the constraint, the projection is the identity. Otherwise, the orthogonal projection onto the closest edge or vertex should be done.

2 Color Diffusion

In this section, we first summarize state-of-the-art methods. We then present a strategy for image colorization based on the total variation minimization. This framework uses some recent state-of-the-art approaches in order to diffuse color strokes on grayscale images. We review some work addressing a coupled total variation with a L2 data-fidelity term. Since this estimator is biased, we then review a debiasing strategy that can be applied on this last model.

2.1 State-of-the-art of color diffusion

Some papers of the literature aim at helping the user to perform manual colorization. This is done by a diffusion of the colors over the gray-scale image by various techniques. The diffusion approaches can also take inspiration from manual colorization to improve the results of other colorization approaches. In this section, we will describe the diffusion techniques proposed in the literature. This chapter is based on the papers [60, 68, 62] that are competitive methods of the literature. Let us remark that there does not exist a perfect diffusion method, all the state-of-the-art approaches having their advantages and drawbacks.

In order to perform manual colorization, a user manually adds color strokes. These are called *scribbles*, and they consist in a set of pixels for which the chrominance channels are defined. Many methods using this process have been proposed. For example, the method of [48] solves an optimization problem for diffusing scribbles on the target image, assuming that the chrominance must have small variations when the luminance does vary a lot. Specifically, the following functional is minimized:

$$H(U) = \sum_{r} \left(U(r) - \sum_{r \sim s} w_{rs} U(s) \right)^2 , \qquad (4)$$

where $r \sim s$ means that pixels r and s are neighbors, and U is a chrominance channel (the same functional is minimized for the channel V). w_{rs} denotes the weights which can be either:

$$w_{rs} \propto e^{(Y(r) - Y(s))^2 / 2\sigma^2} ,$$

or:

$$w_{rs} \propto 1 + \frac{1}{\sigma_r^2} (Y(r) - \mu_r)(Y(s) - \mu_r) ,$$

where μ_r and σ_r denote the mean and the variance of the neighborhood of the pixel r. The two types of weights are more or less sensitive to the variation of contrast. The authors of [53] includes texture similarity in the model of [48] to improve the diffusion process.

The authors of [82] have proposed a simple and fast method using geodesic distance to weight for each pixels the melting of the colors given by the scribbles.

For each pixels of the gray-scale image, the geodesic distance from the scribble is computed with respect to the gradient of the image. Next, a weighted average of the chrominances given by the scribbles is computed. The weights are computed from a function depending on the geodesic distance. This method enables a diffusion of the chrominance on constant parts of the image with respect to a function having similar properties as the inverse function:

- $\lim_{r \to 0} w(r) = \infty$;
- $\lim_{r \to \infty} w(r) = 0$;
- $\lim_{r \to \infty} w(r + r_0)/w(r) = 1$.

Yatziv *et al.* have proposed experimental results with the function $\dfrac{1}{r^b}$ with $1 \leq b \leq 6$.

The authors of [41] have extended this method to textured images by introducing texture descriptors in the diffusion potential.

Some methods are designed as a propagation of the colors from neighbors to neighbors. Some colors are given by strokes drawn by the user. In this way, some of the image pixels are colored. The algorithm then propagates the color to their neighbors with a rule based on the values of the gray-scale image. To this aim, the authors of [33] give an explicit formula for melting the neighbor colors, whereas the ones of [45] provides a modeling based on probabilistic distance transform, and the authors of [42] uses random walks.

It was also proposed to use diffusion through the regularization of non-local graphs. The method proposed by [52] is based on the regularity of the image. This is modeled as a graph, each pixel being represented by a vertex, and each neighborhood relationship by an edge. A local graph is considered, where each edge represents a relationship of 8 neighborhoods. The weight of an edge being inversely proportional to the difference between grey levels, the minimization of an energy depending on these weights (see for example, [49]), enables to diffuse the chrominances on the constant parts of the image. If a non-local graph is designed with a weight which depends on the distance between patches, a set of pixels is considered constant if the patches are similar. Thus, the color of the scribbles is diffused between pixels close in the graph, therefore belonging to similar textures.

Inspired by PDE diffusion scheme [56] some chrominance diffusion including a guidance with Di Zenzo tensor structure computed from gray-scale image was proposed independently by [58] and by [24].

The authors of [69] have proposed a variational approach in chromaticity-brightness color space (see, *e.g.*, [10]) to interpolate missing colors. The *reproducing kernel Hilbert spaces* (RKHS) are used to compute a link between the chromaticity and brightness channels. [38] introduced a variational model with coupling of contours directions. Based on Mumford-Shah type functional, the authors of [39] introduced a novel variational image colorization model. In the following, we present a recent state-of-the-art method based on total variation minimization. This approach enables to combine various strategies of the literature.

2.2 Coupled Total Variation for Image Colorization

In the following we focus on a variational model to denoise the chrominance channels of an image while keeping the luminance unchanged. Similarly to the colorization model of [60], we want to find the minimizer $\hat{u}(c)$ of the denoising functional:

$$\hat{u}(c) = \operatorname{argmin}_{u=(U,V)} \operatorname{TV}_{\mathfrak{C}}(u) + \lambda \int_{\Omega} \|u(x) - c(x)\|^2 \, dx + \chi_{\mathcal{R}}(u), \qquad (5)$$

with

$$\operatorname{TV}_{\mathfrak{C}}(u) = \int_{\Omega} \sqrt{\gamma \|\nabla Y(x)\|^2 + \|\nabla U(x)\|^2 + \|\nabla V(x)\|^2} \, dx, \qquad (6)$$

where Y, U and V are the luminance and chrominance channels. This term is a coupled total variation which enforces the chrominance channels to have a contour at the same location as the luminance one. γ is a parameter which enforces the coupling of the channels. Some others total variation formulations have been proposed to couple the channels, see for instance [40] or [8].

The fidelity data term is a classical L^2 norm between chrominance channels of the unknown u and the data c. For each pixel, the chrominance values live onto the convex polygon denoted by $\mathcal{R}$ and described in Section 1.3. This last assumption ensures that the final solution lies onto the RGB cube, avoiding final truncation that leads to modification of the luminance channel. Model (5) is convex and it can be turned into a saddle-point problem of the form:

$$\min_{u \in \mathbb{R}^2} \max_{z \in \mathbb{R}^6} \frac{\lambda}{2} \|u - c\|^2 + \langle \nabla u | z_{1,\dots,4} \rangle + \langle \gamma \nabla Y | z_{5,\dots,6} \rangle - \chi_{B(0,1)}(z) + \chi_{\mathcal{R}}(u). \qquad (7)$$

The primal-dual algorithm [9] used to compute such saddle-point is recalled in Algorithm 2, where $P_{\mathcal{R}}$ is the orthogonal projection described in Algorithm 1 and $P_{\mathcal{B}}$ is defined as follows for one pixel:

$$P_{\mathcal{B}}(z) = \frac{\left(z_{1,\dots,4}, z_{5,6} - \sigma \nabla Y\right)}{\max\left(1, \left\|z_{1,\dots,4}, z_{5,6} - \sigma \nabla Y\right\|_2\right)}. \qquad (8)$$

Algorithm 2 Minimization of (7).

1: $u^0 = c$
2: $z^0 \leftarrow \nabla u$
3: **for** $n \geq 0$ **do**
4:     $z^{n+1} \leftarrow P_{\mathcal{B}}(z^n + \sigma \nabla \bar{u}^n)$
5:     $u^{n+1} \leftarrow P_{\mathcal{R}} \left(\dfrac{u^n + \tau \left(\operatorname{div}(z^{n+1}) + \lambda c\right)}{1 + \tau \lambda} \right)$
6:     $\bar{u}^{n+1} \leftarrow 2u^{n+1} - u^n$
7: **end for**
8: set $\hat{u}(c) = u^{n+1}$ and $\hat{z} = z^{n+1}$.

The results produced by Algorithm 2 are promising but with a low data parameter λ, the results are drab (see, *e.g.*, [68]).

2.3 Constrained TV-L2 Debiasing Algorithm

In this section we present a debiased algorithm for correcting the loss of colorfulness of the solution given by the optimum of (5).

2.3.1 The CLEAR Method [19]

The CLEAR method [19] can be applied for debiasing estimators $\hat{u}(c)$ obtained as :

$$\hat{u}(c) \in \mathrm{argmin}_{u \in \mathbb{R}^p} \; F(u, c) + G(u), \tag{9}$$

where F is a convex data fidelity term with respect to the data c and G is a convex regularizer. For G being the Total Variation regularization, the estimator $\hat{u}(c)$ is generally computed by an iterative algorithm, and it presents a loss of contrast with respect to the data c. In order to debias this estimator, the CLEAR method refits the data c with respect to some structural information contained in the biased estimator $\hat{u}$. This information is encoded by the Jacobian of the biased estimator with respect to the data c:

$$J_{\hat{u}}(c)d = \lim_{\varepsilon \to 0} \frac{\hat{u}(c + \varepsilon d) - \hat{u}(c)}{\varepsilon}. \tag{10}$$

For instance, when G is the anisotropic TV regularization, the Jacobian contains the information concerning the support of the solution $\hat{u}$, on which a projection of the data can be computed.

In the general case, the CLEAR method relies on the *refitting estimator* $\mathcal{R}_{\hat{u}}(c)$ of the data c from the biased estimation $\hat{u}(c)$:

$$\mathcal{R}_{\hat{u}}(c) \in \mathrm{argmin}_{h \in \mathcal{H}} \; \|h(c) - c\|_2^2 \tag{11}$$

where $\mathcal{H}$ is defined as the set of maps $h : \mathbb{R}^n \to \mathbb{R}^p$ satisfying, $\forall c \in \mathbb{R}^n$:

$$h(c) = \hat{u}(c) + \rho J_{\hat{u}(c)}(c - \hat{u}(c)), \text{ with } \rho \in \mathbb{R}. \tag{12}$$

A closed formula for ρ can be given:

$$\rho = \begin{cases} \dfrac{\langle J_{\hat{u}(c)}(\delta) | \delta \rangle}{\|J_{\hat{u}(c)}(\delta)\|_2^2} & \text{if } J_{\hat{u}(c)}(\delta) \neq 0 \\ 1 & \text{otherwise.} \end{cases}, \tag{13}$$

where $\delta = c - \hat{u}(c)$. In practice, the global value ρ allows to recover most of the bias in the whole image domain.

An algorithm is then proposed in [19] to compute the numerical value of $J_{\hat{u}(c)}(c - \hat{u}(c))$. The process is based on the differentiation of the algorithm providing $\hat{u}(c)$.

It is important to notice that the CLEAR method applies well for estimators obtained from the resolution of unconstrained minimization problems of the form (9). Nevertheless, it is not adapted to the denoising problem (5) that contains an additional constraint $\chi_{\mathcal{R}}(u)$ as CLEAR may violate the constraint.

2.3.2 Direct Extension of CLEAR to Constrained Problems

Extending the CLEAR method to the constrained model (5) requires to take the constraint into account in the axioms of the refitting model (11). The main difference with the original model is the addition of the constraint $\chi_{\mathcal{R}}(u)$. We can first notice that the refitting axioms $h(c) = Ac + b$ for some $A \in \mathbb{R}^{p \times n}$, $b \in \mathbb{R}^p$ and $J_h(c) = \rho J_{\hat{u}}(c)$ for some $\rho \in \mathbb{R}$ are in line with the introduction of the constraint. In particular, the definition of the Jacobian $J_{\hat{u}}$ in Equation (10) remains valid with the constraint, since $\hat{u}(c)$ and $\hat{u}(c + \varepsilon d)$ are still in the closed convex $\mathcal{R}$. The computation of the ρ parameter in Equation (13) may nevertheless produce, from Equation (12), an estimation out of the constraint, that has to be post-processed. This points out the main difference between the constrained and the unconstrained debiased estimator.

In [19], the value of ρ is computed from the minimization of a map from $\mathbb{R}$ to $\mathbb{R}$ defined as follows :

$$\rho \mapsto \| \left(I_d - \rho J_{\hat{u}(c)} \right) (\hat{u}(c) - c) \|_2^2. \tag{14}$$

In the case of the constrained problem, the function to be minimized is written as :

$$\rho \mapsto \|\hat{u}(c) + \rho J_{\hat{u}(c)} (c - \hat{u}(c)) - c\|_2^2 + \chi_{\mathcal{R}}(\hat{u}(c) + \rho J_{\hat{u}(c)} (c - \hat{u}(c))). \tag{15}$$

Let us denote by ρ the value defined in Equation (13). In the case when the constraint is fulfilled, *i.e.*, when $\hat{u}(c) + \rho J_{\hat{u}(c)} (c - \hat{u}(c)) \in \mathcal{R}$, then, the minimum of (15) is reached with ρ.

If not, since function (15) is convex, it is possible to compute explicitly the minimizer. The value $\rho = 0$ is in the domain of the functional because $\hat{u}(c) \in \mathcal{R}$. The idea is to find the maximum value of ρ such that $\hat{u}(c) + \rho J_{\hat{u}(c)} \delta \in \mathcal{R}$. In this case, since $\mathcal{R}$ is a convex polygon, this computation can be done with a Ray-Tracing algorithm [80]. To this aim, we can parametrize the segment $[\hat{u}(c), \hat{u}(c) + \rho J_{\hat{u}(c)} \delta]$:

$$\tilde{\rho} = \max_{t \in [0,1]} t\rho \text{ such that } \hat{u}(c) + t\rho J_{\hat{u}(c)} (c - \hat{u}(c)) \in \mathcal{R}. \tag{16}$$

Equation (16) can thus be directly solved by the maximum value t such that $\hat{u}(c) + t\rho J_{\hat{u}(c)} (c - \hat{u}(c))$ intersects the border of $\mathcal{R}$.

2.3.3 Direct Debiasing Process

Let us summarize the refitting algorithm designed for model (5). The first step consists in computing a solution of (5) with Algorithm 2. This iterative algorithm provides at convergence a first biased solution $\hat{u}(c)$ and its dual variable $\hat{z}$. Once this solution has been computed, the differentiated algorithm, presented in Algorithm 3, is applied in the direction $\delta = c - \hat{u}(c)$. This algorithm requires the definition of the operator $\Pi_{\hat{z}}(\tilde{z})$ which is the linearized version of the projection $P_{\mathcal{B}}$ around $\hat{z}$, and which reads [19]:

$$
\Pi_{\hat{z}}(\tilde{z}) = \begin{cases} \tilde{z} & \text{if } \|\hat{z}\| < 1 \\ \dfrac{1}{\|\hat{z}\|} \left(\tilde{z} - \dfrac{\langle \hat{z} | \tilde{z} \rangle}{\|\hat{z}\|^2} \hat{z} \right) & \text{otherwise.} \end{cases} \tag{17}
$$

Finally, the Ray-Tracing is applied to obtain $\tilde{\rho}$ and get the debiased solution as $\hat{u}(c) + \tilde{\rho} J_{\hat{u}(c)}(c - \hat{u}(c))$.

Algorithm 3 Differentiation of Algorithm 2 for computing $J_{\hat{u}(c)}\delta$ from $(\hat{u}(c), \hat{z})$

1: $\tilde{u}^0 = \delta, \overline{\tilde{u}}^0 = \delta$
2: $\tilde{z}^0 \leftarrow \nabla \tilde{u}$
3: **for** $n \geq 0$ **do**
4: $\quad \tilde{z}^{n+1} \leftarrow \Pi_{\hat{z}} \left(\tilde{z}^n + \sigma \nabla \overline{\tilde{u}}^n \right)$
5: $\quad \tilde{u}^{n+1} \leftarrow \dfrac{\tilde{u}^n + \tau \left(\mathrm{div}(\tilde{z}^{n+1}) + \lambda \delta \right)}{1 + \tau \lambda}$
6: $\quad \overline{\tilde{u}}^{n+1} \leftarrow 2\tilde{u}^{n+1} - \tilde{u}^n$
7: **end for**
8: $J_{\hat{u}(c)}\delta = \tilde{u}^{n+1}$.

Unfortunately, this direct approach does not lead to valuable results on general cases. Indeed, if for one particular pixel the solution $\hat{u}(c)$ is saturated, and if the debiased solution is out of $\mathcal{R}$, then $\tilde{\rho} = 0$ is the unique global ρ satisfying $\hat{u}(c) + \rho J_{\hat{u}(c)}(c - \hat{u}(c)) \in \mathcal{R}$. Thus, the debiased solution is equal to the biased one, and the debiasing algorithm has no action.

In the next section, we propose a model with an adaptive ρ parameter, depending on the pixel, to tackle this saturated values issue.

2.3.4 Adaptive Debiasing Model for Constrained Problems

For a pixel ω such that $\hat{u}(c)_\omega + \rho J_{\hat{u}(c),\omega} (c_\omega - \hat{u}(c)_\omega)$ fulfills the constraint, ρ is the best value to refit the model according to the hypothesis of model (11). Here, $J_{\hat{u}(c),\omega}$ denotes the value of $J_{\hat{u}(c)}$ in pixel ω.

On the other hand, if for a pixel ω, the value of $\hat{u}(c)_\omega$ and $J_{\hat{u}(c),\omega} (c_\omega - \hat{u}(c)_\omega)$ are such that $\hat{u}(c)_\omega + \rho J_{\hat{u}(c),\omega} (c_\omega - \hat{u}(c)_\omega) \notin \mathcal{R}$, the ρ value has to be adapted. Thus, let

us define for a pixel ω the adapted $\tilde{\rho}_\omega$ as follows:

$$\tilde{\rho}_\omega = \max_{t_\omega \in [0,1]} t_\omega \rho \text{ such that } \hat{u}(c)_\omega + t_\omega \rho J_{\hat{u}(c),\omega} (c_\omega - \hat{u}(c)_\omega) \in \mathcal{R}. \qquad (18)$$

The constrained refitting model is then defined pixel-wise as:

$$R^{\mathcal{R}}_{\hat{u}}(c) = \hat{u}(c)_\omega + \tilde{\rho}_\omega J_{\hat{u}(c),\omega} (c_\omega - \hat{u}(c)_\omega) \qquad (19)$$

This definition ensures that the debiased estimation fulfill the constraint. More-over, if the debiasing method of [19] produces an estimation that fulfills the constraint, this solution is retained. Notice however that the CLEAR hypothesis $J_h(c) = \rho J_{\hat{u}}(c)$ for some $\rho \in \mathbb{R}$ in model (11) is not fulfilled anymore. In numerical experiments, for most pixels, the values of $\tilde{\rho}_\omega$ computed with this method are the same as with Model (11).

As illustrated by Figure 3, such a local debiasing strategy realizes an oblique projection onto $\mathcal{R}$.

2.3.5 Computation of the Oblique Projection

In [68], an algorithm used to compute the oblique projection when the constraint is the chrominance set for a particular value of luminance (see, *e.g.*, Section 1.3) is proposed. To simplify the notation, the problem is considered for a single pixel ω and one set $u := \hat{u}(c)_\omega$, $c := J_{\hat{u}(c),\omega} (c_\omega - \hat{u}(c)_\omega)$ and $\rho \in \mathbb{R}$ computed by the algorithm of [19].

For $u + \rho c \notin \mathcal{R}$, the maximum value of $t \in [0,1]$ such that $u + t\rho c \in \mathcal{R}$ is computed. Since $u \in \mathcal{R}$, thus if $u + \rho c \notin \mathcal{R}$, the segment $[u, u + \rho c]$ crosses one edge of the polygon.

One considers this problem by testing it into the RGB space. Indeed, the edges in the chrominance space correspond to edges in the RGB one, and the intersections between them correspond to intersections in the RGB space. In RGB, the problem of finding the intersection between an edge and the polygon is reduced to computing the intersection between the edge and the cube faces because the edges of the polygon are included in the cube by construction (see, *e.g.*, Figure 1(a)).

The transformation of the chrominance values $u = (U, V)$ to the RGB space with the luminance value Y is denoted by $T_Y(u)$. From the expression of the standard transformation from RGB to YUV, we have $T_Y(u) = Y (1, 1, 1)^t + L(U, V)$ with L a linear function. The following equalities come:

$$\begin{aligned} T_Y(u + \rho c) &= Y (1, 1, 1)^t + L(u + \rho c) \\ &= Y (1, 1, 1)^t + L(u) + \rho L(c) \\ &= T_Y(u) + \rho T_Y(c) - \rho Y (1, 1, 1)^t. \end{aligned} \qquad (20)$$

It is required to compute $\tilde{\rho}$ such that $T_Y(u + \tilde{\rho}c)$ is at the boundary of the RGB cube. To this aim, the 6 different values $\tilde{\rho}_c^v$ with $c \in \{R, G, B\}$ and $v \in \{0, 255\}$
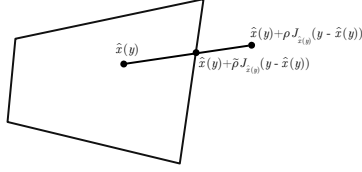
Fig. 3 The refitting of the method of [19] may be out of the constraint. An oblique projection onto this constraint is able to respect most of hypotheses of the Model (11) while fulfilling the constraint.
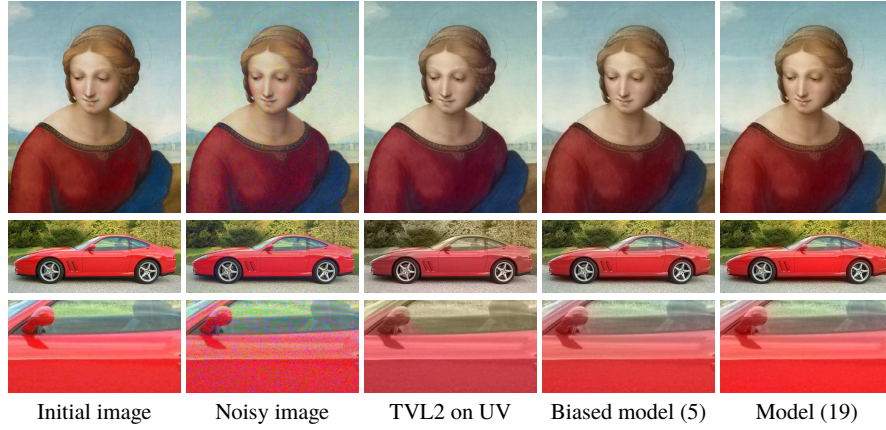

| Initial image | Noisy image | TVL2 on UV | Biased model (5) | Model (19) |

Fig. 4 Results of chrominance channels with a TVL2 model on chrominance, with the biased method and with the unbiased method. The debiasing algorithm produces more colorfull results.

corresponding to the cases where the 3 coordinates of $T_Y(u + \tilde{\rho}c)$ are equal to 0 or 255 are computed. For instance, if the first coordinate R of $T_Y(u + \tilde{\rho}c)$ is equal to 255, we have:

$$T_Y(u + \tilde{\rho}_R^{255}c)_R = 255 \tag{21}$$

$$T_Y(u)_1 + \tilde{\rho}_R^{255}T_Y(c)_R - \tilde{\rho}_R^{255}Y = 255. \tag{22}$$

so that

$$\tilde{\rho}_R^{255} = \frac{255 - T_Y(u)_R}{T_Y(c)_R - Y}. \tag{23}$$

For each of the six values $\tilde{\rho}_c^v$ computed as in Equation (23), one can compute $t_c^v = \frac{\tilde{\rho}_c^v}{\rho}$. The values t_c^v that are between 0 and 1 correspond to an intersection of the segment $[u, u + \rho c]$ with the boundaries of $\mathcal{R}$. One finally takes $t^* = \min_{t_c^v \in [0;1]} t_c^v$ and the result of Equation (18) is given by $t^*\rho$.

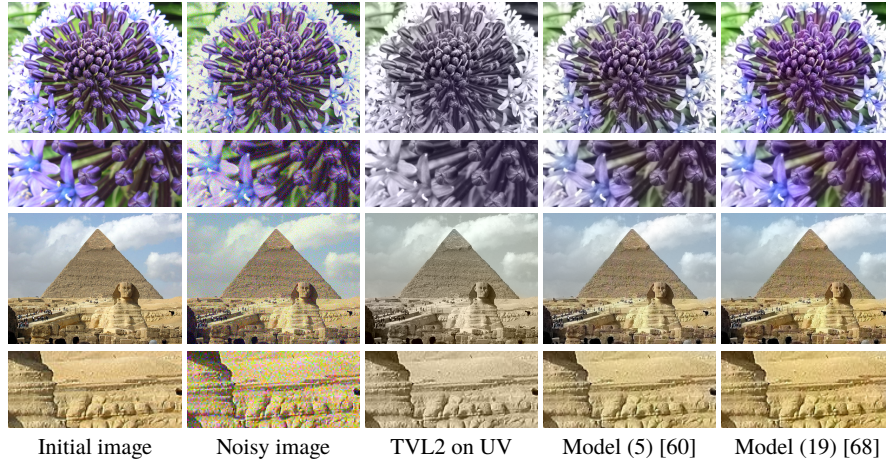| Initial image | Noisy image | TVL2 on UV | Model (5) [60] | Model (19) [68] |

Fig. 5 The advantage of the coupled total variation (5) on the TVL2 model has been shown in [60]. In [68], it is refined in a better colorfulness-preserving model.

To summarize, to design a suitable variational model for image colorization, the three mains ingredients are the coupled total variation, the orthogonal projection onto the range of the problem, and the debiasing algorithm. This variational model is a basis for image colorization in many paradigms. In the next sections, some concrete cases of application of this model are presented in the case of exemplar-based approaches or coupled with manual techniques or CNN-based framework.

3 Examplar-based Colorization

The manual methods enable the user to chose the color in each pixel of the image. Nevertheless, their main drawback is the tedious work needed for complex scenes, for instance with textures. In exemplar-based image colorization methods, the color information is provided by a color image called *source image*. The grayscale image to colorize is called *target image*. This color image can be chosen by the user or automatically provided from a database with an indexation algorithm.

The results available in this chapter are based on [67, 60, 62] which are among the most recent methods in patch-based colorization and on [57] which is the current most competitive method for exemplar-based colorization of face images.

In order to transfer the colors from the source image to the target one, three concepts have been proposed in the literature. One of them is based on geometry, the two others are based on texture similarities. The first one is specifically well adapted to faces colorization. In the first part of this section, we will review the work of [57] which is the current most competitive method for exemplar-based colorization of face images. Next, we will present an overview of segmentation-based approaches

which use the texture similarities on segmented parts of the images to transfer colors. Finally, we present patch-based technique which avoid the requirement of an efficient segmentation method and which can be coupled with a variational model.

3.1 Morphing-based Approach

In this section, we describe the model of [57]. The authors compute the morphing map between the two gray-scale images I_{temp} and I_{tar} with a model inspired by [4]. This results in the deformation sequence φ which produces the resulting map Φ from the template image to the target one. Due to the discretization of the images, the map Φ is defined, for images of size $n \times m$, on the discrete grid $\mathcal{G} := \{1...n\} \times \{1...m\}$:

$$\Phi : \mathcal{G} \rightarrow [1, n] \times [1, m], \quad x \mapsto \Phi(x), \tag{24}$$

where $\Phi(x)$ is the position in the source image which corresponds to the pixel $x \in \mathcal{G}$ in the target image. Now we colorize the target image by computing its chrominance channels, denoted by $(U_{\text{tar}}(x), V_{\text{tar}}(x))$ at position x as

$$\big(U_{\text{tar}}(x), V_{\text{tar}}(x)\big) := \big(U(\Phi(x)), V(\Phi(x))\big). \tag{25}$$

The chrominance channels of the target image are defined on the image grid $\mathcal{G}$, but usually $\Phi(x) \notin \mathcal{G}$. Therefore the values of the chrominance channels at $\Phi(x)$ have to be computed by interpolation. In the algorithm, bilinear interpolation is simply used, which is defined for $\Phi(x) = (p, q)$ with $(p, q) \in [i, i + 1] \times [j, j + 1]$, $(i, j) \in \{1, \ldots, m - 1\} \times \{1, \ldots, n - 1\}$ by

$$U(\Phi(x)) = U(p, q) := (i + 1 - p, p - i) \begin{pmatrix} U(i, j) & U(i, j + 1) \\ U(i + 1, j) & U(i + 1, j + 1) \end{pmatrix} \begin{pmatrix} j + 1 - q \\ q - j \end{pmatrix}. \tag{26}$$

Finally, a colorized RGB image is computed from its luminance $I_{\text{tar}} = Y_{\text{tar}}$ and the chrominance channels.

Figure 6 summarizes the color transfer method.

The technique proposed in [57] is adapted to faces. To address the problem of colorization of textured images geometric similarities are nor reliable. Texture similarities have to be obviously compared. Such approaches are reviewed in the next sections.

3.2 Segmentation-based Techniques

In order to transfer the colors from the source image to the target one, a lot of approaches are based on an image segmentation techniques in order to compare
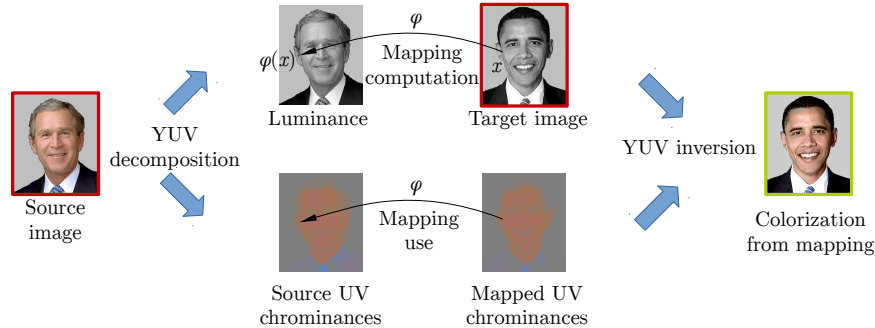
Fig. 6 Overview of the color transfer. The mapping φ is computed from a model inspired by [4] between the luminance channel of the source image and the target one. From this map, the chrominances of the source image are mapped. Finally, from these chrominances and the target image the colorization result is computed.

the statistical attributes of the textures. For instance, the authors of [35] proposed to compute the best correspondence between the target image and some segmented parts of the source image. From these correspondences, some *micro-scribbles* are drawn of the target image from the source image and the color strokes are then propagated by the diffusion technique in [48]. In [76], the author used a segmentation approach to colorize images of old cartoons. The method of [30] extracts various descriptors from superpixel segmentation (see, *e.g.*, [70, 2]) from target image and matches them with the ones of the target image with these various descriptors (SURF, mean, standard-deviation, Gabor filters, etc). The method draws hence one scribble for each superpixel from this matching. The final color is computed from the optimization of a criterion which favors a spatial consistency of the colors as done in [48]. A similar approach has been proposed in [44].

The efficiency of these methods depend on the preliminary segmentation of the images. In the next section, we propose an overview of patch-based techniques which avoid this preliminary step.

3.3 Patch-Based Methods

The first patch-based method for image colorization is the one proposed by [79], which is widely inspired by the texture synthesis algorithm introduced by the authors of [25]. It is based on the patch similarities in the colorization process.

First, a luminance remapping (see, *e.g.*, [32]) is done as a first step: in order to make the luminance values more comparable between the source image and the target one, an affine mapping is used on the luminance of the source image in order to better match the histogram of the luminance channel. Indeed, the range of the luminance channels could be different and the comparison of these channels could be senseless.
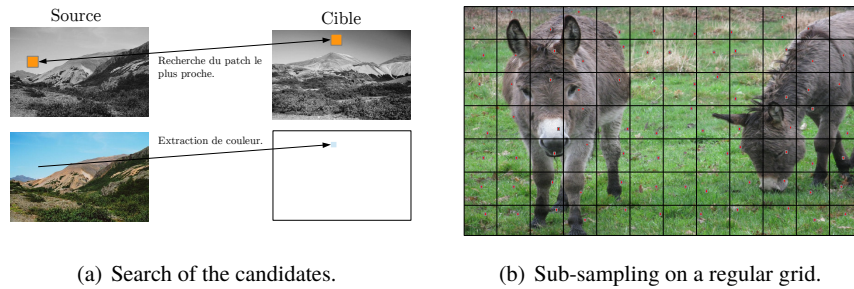
(a) Search of the candidates. (b) Sub-sampling on a regular grid.

Fig. 7 For each pixel of the target image, the method compares the patch centered on the pixel with ones of the gray-scale version of the source. Next, the method retains the color of the central pixel of the closest patch (see (a)). To speed up the algorithm, the search is not performed among all pixels, but only on a sub-sampling (see (b)).

Next, for each pixel of the target image, the algorithm compare the patch centered in this pixel with a set of patches extracted from the luminance channel of the source image. Once the closest patch computed, the chrominance values of the pixel at the center of the patch of the source image are extracted and provided to the considered pixel in the target image (see, *e.g.*, Figure 7). In combination with the luminance of the target image and the chrominance values extracted from the source image, a RGB color is given.

The set of the reference patches extracted from the source image is a subset of patches randomly chosen in this way: the image is divided within a regular grid and one pixel is chosen randomly on each part of this grid (see, *e.g.*, Figure 7(b)).

The authors of [22] proposed an improvement which speeds-up the patch research with a tree-clustering algorithm inspired from [78]. Next, the authors of [15] proposed an improvement based on a Bayesian approach.

The patch based approaches suffers from two drawbacks which are the difficulty to chose a reliable metric to compare patches, and the spatial coherency in the border of two areas with different textures. We will see in the following how to overcome these limitations.

The patch-based approaches need some metrics in order to compare patches. Unfortunately, there does not exist any perfect metric, each of them having its advantages and drawbacks. In most computer vision problems, the algorithms have to distinguish objects or textures with the same accuracy and the same sensitivity as human visual system. Metric for textures comparison are based on numerical data. The link between this data and the human visual system is done by features that are vectors which describes the local statistic of the image.

The most simple metrics are based on the mean or the standard-deviation of the patches, whereas some others use histograms, Fourier transform, SURF features [3], structure tensors, co-variance matrices, Gabor features, etc.

Based on various patch metrics, it is thus possible to get many exemplar-based colorization results. In the following, we focus on the fusion of such results to obtain only one final result.

Experimentally, the authors of [5] have used he following descriptors:

- the standard deviation on 5×5 and 3×3 patches;
- the spectrum amplitude (FFT) on 7×7, 9×9 and 11×11 patches;
- difference in L^1 norm of the cumulative histograms on 7×7, 9×9 and 11×11 patches.

These descriptors are used by the authors of [5] to extract eight color candidates for each pixel in the same way as done in [79]. For each metric, the method retains the pixel of the source image corresponding to the closest patch with respect to this metric. After this step, for each pixel of the target image, eight pixels of the source image can match. To summarize, each pixel having its luminance and eight chrominance values coming from the matched pixels (see, *e.g.*, Figure 8), eight colors are available, called *color candidates*. In the work of [5], the colors are used directly, whereas in [61] an oblique projection in the RGB color space is proposed in order to avoid some artificial modification of the Hue due to gamut problems.

Some other metrics could be used. For instance, whereas the method of [12] is not based on patch decomposition, it uses a local representation with SURF descriptors to predict color in each pixel. Let us mention that this method also requires numerous and complex steps.
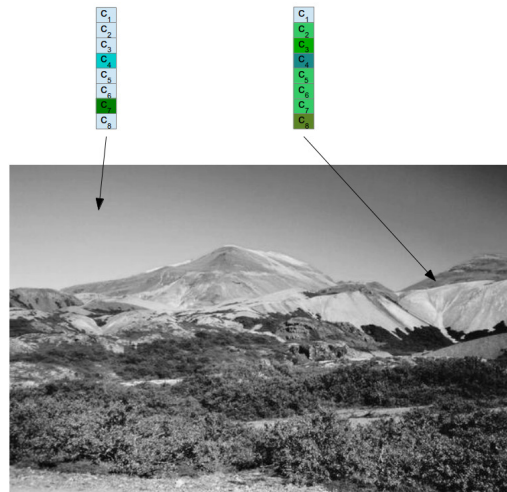

Fig. 8 Some methods of the literature begin with the search of C candidates per pixel (here $C = 8$).

With multiple color candidates coming from various descriptors, a choice has to be done among them. In the following we will consider a generic number of color

candidates denoted by C. The aim of the methods describe hereinafter consist in the selection of one of the color candidate. Let us notice that the choice of an ideal metric based on metric learning has been proposed in [63] but with rather worst results than state-of-the-art due to a lack of spatial regularization of the results. In order to retain only one color per pixels, the authors of [5] proposed to compute a median of the candidates based on an order between them computed with a standard PCA of the set of colors. This PCA is required because there is no natural order in the RGB space of colors. The method of [51, 50] provides an order in the set of colors, but it requires some neighborhood information which is not available here.

Let us remark that the method of [5] does not use the spatial regularization or spatial coherency of the color to chose a color candidate. The authors of [37] proposed an extension to exemplar-based colorization of [39] with color inference based on patch descriptors (DFT, etc). A variational method similar to [60] is proposed to regularize the final results.

3.4 A Variational Model for Image Colorization with Channels Coupling

In [60], the authors have proposed a functional that selects a color among candidates extracted from a patch-based method, inspired by the method of [6] in order to tackle some issues (numerical cost of numerical scheme, halo effects, etc). Assuming that C candidates are available in each pixel of a domain Ω and assuming that two chrominance channels are available for each candidate. Let us denote for each pixel at position x the i-th candidate by $c_i(x)$, $u(x) = (U(x), V(x))$ stands for chrominances to compute, and $w(x) = \{w_i(x)\}$ with $i = 1, \ldots, C$ for the candidate weights. Let us minimize the following functional with respect to (u, w):

$$F(u, w) := TV_{\mathfrak{C}}(u) + \frac{\lambda}{2} \int_{\Omega} \sum_{i=1}^{C} w_i \|u(x) - c_i(x)\|_2^2 \, dx + \chi_{\mathcal{R}}(u(x)) + \chi_{\Delta}(w(x)) \,.$$

(27)

The central part of this model is based on the term

$$\int_{\Omega} \sum_{i=1}^{C} w_i(x) \|u(x) - c_i(x)\|_2^2 \, dx.$$

(28)

This term is a weighted average of some L^2 norms with respect to the candidates c_i. The weights w_i can be seen as a probability distribution of the c_i. For instance, if $w_1 = 1$ and $w_i = 0$ for $2 \leq i \leq C$, the minimum of F with respect to u is equal to the minimization of

$$TV_{\mathfrak{C}}(u) + \frac{\lambda}{2} \int_{\Omega} \|u(x) - c_1(x)\|_2^2 \, dx + \chi_{\mathcal{R}}(u(x)).$$

(29)

To simplify the notations, the dependence of each value to the position x of the current pixel will be removed in the following. For instance, the second term of (27) will be denoted by $\int_{\Omega} \sum_{i=1}^{C} w_i \|u - c_i\|_2^2 \, dx$.

This model is a classical one with a fidelity-data term $\int_{\Omega} \sum_{i=1}^{C} w_i \|u - c_i\|_2^2$ and a regularization term $TV_{\mathfrak{S}}(u)$ defined in Equation (6). Since the first step of the method extracts many candidates, we propose averaging the fidelity-data term issued from each candidate. This average is weighted by w_i. Thus, the term

$$\int_{\Omega} \sum_{i=1}^{C} w_i \|u - c_i\|_2^2 \tag{30}$$

connects the candidate color c_i to the color u that will be retained. The minimum of this term with respect to u is reached when u is equal to the weighted average of candidates c_i.

Since the average is weighted by w_i, these weights are constrained to be onto the probability simplex. This constraint is formalized by $\chi_{\Delta}(w)$ whose value is 0 if $w \in \Delta$ and $+\infty$ otherwise, with Δ defined as:

$$\Delta := \left\{ (w_1, \cdots, w_C) \text{ s.t. } 0 \leq w_i \leq 1 \text{ and } \sum_{i=1}^{C} w_i = 1 \right\}. \tag{31}$$

In order to compute a suitable solution for the problem in (27), authors of [60] propose a primal-dual algorithm with alternating minimization of the terms depending of w. They also proposed numerical experiments showing the convergence of their algorithm. Let us note that this recent reference shows that the convergence of such numerical schemes can be demonstrated after smoothing of the total variation term. Among all the numerical schemes proposed in the references [60, 77], we choose the methodology having the best convergence rate as well as a convergence proof. This scheme is given in Algorithm 2 in [77]. This algorithm is a block coordinate forward backward algorithm. To increase the speed-up of the convergence, Algorithm 2 of [77] is initialized with the result of 500 iterations of the primal-dual algorithm of [60]. Whereas this algorithm has no guaranty of convergence, the authors of [77] have experimentally observed that it numerically converges faster.

Unfortunately, the functional (27) is highly non-convex and it contains many critical points. More precisely, the functional is convex with respect to u with fixed w and reversely, it is convex with respect to w for fixed u. Nevertheless, the functional is not convex with respect to the joint variables (u, w). Thus, even if the numerical scheme would converge to a local minimum, the solution of the problem depends on the initialization.

The dependence to the initialization implies an influence of the source image for exemplar-based colorization, and it does not enable a fully automatic image colorization within this paradigm. In the next section, we will show how the colorization from datasets can be used to tackle this last limitation.

4 Colorization from Dataset

The third colorization approach uses some large image databases [83]. Neural networks (Convolutional Neural Networks, Generative Adversarial Networks, Autoencoder, Recursive Neural Networks) have also been used successfully leading to a significant number of recent contributions. The survey proposed in this section is based on the paper [54]. This literature can be divided into two categories of methods. The first evaluates the statistical distribution of colors for each pixel [83, 71, 14]. The network computes, for each pixel of the gray-scale image, the probability distribution of the possible colors. The second takes a grayscale image as input and provides a color image as output, mostly in the form of chrominance channels [34, 47, 7, 36, 21, 29, 31, 75]. Some methods use a mixture of both (*e.g.*, [84]).

Both techniques require image resizing, that is either done by deconvolution layers or performed *a posteriori* with standard interpolation techniques.

| Target (input) | Result of [83] | Result of [54]. |

Fig. 9 Example of halo effects produced by the method of [83]. Based on a variational model, the approach of [54] is able to remove such artifacts.

In the case of [83], the network computes a probability distribution of the color on a down-sampled version of the original image. The choice of a color in each pixel at high resolution is made by linear interpolation without taking into account the grayscale image. Hence, the contours of chrominance and luminance may be not aligned, producing halo effects. Figure 9 shows some grey halo effects at the bottom of the cat that are visible on the red part, near the tail. On the other hand, in comparison to the others approaches of the state-of-the-art, the method of [83], produces images which are shinier.

Below, the CNN described in [83] is presented in detail. The method of [83] is based on a discretization of the CIE Lab color space into C=313 colors. This number of reference colors comes from the intersection gamut of the RGB color space and the discretization of the Lab space. The authors designed a CNN based on a VGG network [73] in order to compute a statistical distribution of the C colors in each pixel. The input of the network is the L lightness channel of the Lab transform of an image of size 256×256. The output is a distribution of probability over a set of 313 couples of a, b chrominance values for each pixel of a 64×64 size image. The quantification of the color space in 313 colors is computed from two assumptions. First, the colors are regularly spaced onto the CIE Lab color space. On this color

space two colors are close with respect to the Euclidean norm when the human visual system feels them close. The second assumption that rules the set of colors is the respect of the RGB gamut. The colors have to be displayable onto a standard screen.

To train this CNN, the database ImageNet [20] is used without the gray-scale images. The images are resized at size 256×256 and then transformed into the CIE Lab colorspace. The images are then resized at size 64×64 to compute the a and b channels. The loss-function used is the cross-entropy between the luminance (a, b) of the training image and the distribution over the 313 original colors. Let us denote by Δ the probability simplex in C=313 dimensions.

Denoting by $(\hat{w}_i(x))_{i=1..C} \in \Delta^N$ the probability distribution of dimension C in the N pixels of the 64×64 image (over a domain Ω), and denoting by $(w_i(x))$ the ground truth distribution computed with a soft-encoding scheme (see [83] for details), the loss-function is given by:

$$L(\hat{w}, w) = -\sum_{x \in \Omega} \sum_{i=1}^{C} w_i(x) \log(\hat{w}_i(x)). \tag{32}$$

The forward propagation in the network provides a probability distribution over the C colors. In order to compute a colorization result, a choice among all these colors has to be performed. Basically, the authors of [83] proposed an annealed-mean in each pixel, independently. After that, a resizing of the (a, b) channels at original size is done and recombined with brightness channel to obtain the color image.

Nevertheless, this recombination is done without taking into account any spatial consideration. In the next section, we will describe some approaches that couple some previously described algorithms.

5 Coupled Approaches

Nor the exemplar-based methods, nor the manual techniques, nor the deep learning approaches are able to colorize images without some defects. All of them having drawbacks or advantages, we propose to describe some coupling approaches that rely on different types of methods in the literature. First, a framework to couple exemplar-based approach and manual colorization is described. A coupling of variational method with deep learning is then recalled.

5.1 Coupling Manual Approach with Exemplar-based Colorization

A method can be considered interactive when the user can influence the result of the colorization process. Nevertheless, the interactivity can be difficult to reach. Indeed, if a method computes a result with a too long delay, the user cannot stand to an intermediary result in order to see the influence of his intervention. The results and

the survey proposed in this section is based on the papers [62, 60] which have led to a software [65].

Some of the exemplar-based methods enable some interaction with the user. For instance, the *swatches* approach of [79] in which the user distinguishes some parts of the image by drawing some rectangles on the source and target images where the textures are similar. The method then colorizes some parts of the target image with the specified parts of the source image. Finally, the method computes a solution for all the remaining uncolored pixels of the image based on the already colorized parts. The advantage of this framework is that the user can easily distinguish or associate the textures of the different images, that is difficult with an automatic method. At the opposite, the exemplar-based method is reliable to well colorize an image from its own parts, because the textures are more similar. With this method, a contextual information is added.

The framework of [16] exploits the huge quantity of data available on the Internet. Nevertheless, the user has to manually segment and label the objects of the target image. Next, for each labeled objects, the images with the same label are found on the Internet and used as source images. The image research is based on superpixel extraction [17] as well as graph-based optimization.

In the work of [23], the scribbles are automatically generated and the user is invited to associate a color to each scribble. Then, the phases of the wavelet in the quaternion space are computed in order to propagate the colors along the lines of equal phase. Indeed, the wavelets in quaternion space are a measure of contours.

The method proposed in [62] consists in a combination of the method of [5] and the one of [82]. The approach uses a GPU implementation to compute a solution of model (27) that enables to colorize an image of size 370×600 in approximately 1 sec. This computation time enables an extension of the exemplar based approach of [67] by including interaction with the user, that leads to a software for colorization [65].

The scribbles can be given in advance or added step-by-step by the user. When a source image is added, the first step consists in the extraction of C candidates as in Section 3.3 and the corresponding weights are initialed with the value $w = 1/C$.

The information given by the scribbles influences the weights and the candidate number. More precisely, for each pixel of the image, a new candidate is added for each scribble. When a candidate is introduced, its weight is initialized for the minimization process with a value depending on the geodesic distance in a similar way as [82].

The geodesic distance, denoted by D, is computed with the Fast Marching algorithm [72] with a potential equal to $\left(0.001 + \|\nabla u\|_2^2\right)^{-4}$ given by [11]. D is normalized to get values between 0 and 1. The implementation of [59] can be used to compute it.

The pixels having a low geodesic distance from a scribble get its color, whereas those having a high geodesic distance are not influenced by the user intervention. The w variable is composed by concatenation of uniform weights for the color candidates coming from the source image with the patch extraction and the weight coming from the geodesic distance. The values are then projected onto the probability simplex

Δ with the algorithm of [15]. The u variable is initialized with $\sum_i w_i c_i$ and the functional (27) is minimized using this initialization.

In Figure 10, we show a first example of colorization using both manual and exemplar-based approach. Figures 10(a) and 10(b) show source and target images. Figure 10(c) corresponds to exemplar-based colorization done without manual scribble. In this first result, the sky is not suitably colorized since it appears brown instead of blue, as well as the door in ruins. Moreover, some blue blotches appear on the floor. Figure 10(d) shows that the corrections done by the user by adding three scribbles
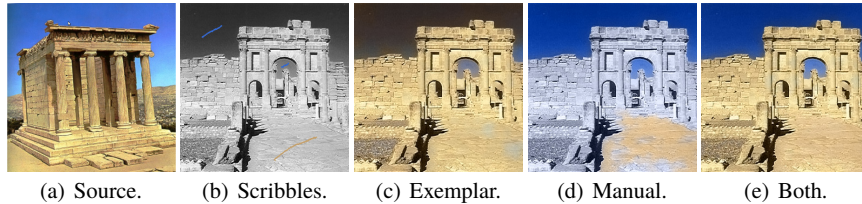

(a) Source.   (b) Scribbles.   (c) Exemplar.   (d) Manual.   (e) Both.

Fig. 10 Colorization using manual and exemplar-based approach (a) Source image, (b) Target image with 3 scribbles, (c) exemplar-based colorization, (d) manual colorization, (e) Both.

on the exemplar-based result (Figure 10(c)). Figure 10(e) illustrates the advantage of the combination of the methods. Indeed, the work provided by the user is lower than full manual colorization. It also shows that Model (27) is able to enhance contours.

Figure 11 shows additional results and illustrate the advantage to use the joint model instead of using only source image (fourth column), or only scribbles (fifth column). Colorization results of the last column in Figure 11 are visually better than the ones computed from only one information source. This experiment shows also that old photographies and faces are difficult to colorize with exemplar-based approaches since they require more scribbles. This statement has been done in [13]. Indeed, old pictures contain a lot of noise and textures. Face image contains smooth parts, for instance skin or background, with no textures. This kind of images is hard to colorize with assumption of texture similarities. Nevertheless, it is possible to compute suitable result with the joint method, as well as morphing-based approach presented in Section 3.1. Let us remark that the scribbles given by the user have naturally a local influence, but this influence can be also considered as global. For instance, on the last row of Figure 11, the blue scribble in the arch also improves the color of the sky in the left-hand part of the image.

5.2 Coupling CNN with a Variational Approach

In the following, we recall the results given in the paper [54] which consists into a coupling between a variational approach with the output of the CNN of [83]. Next, we perform numerical comparisons with the original CNN approach of [83].
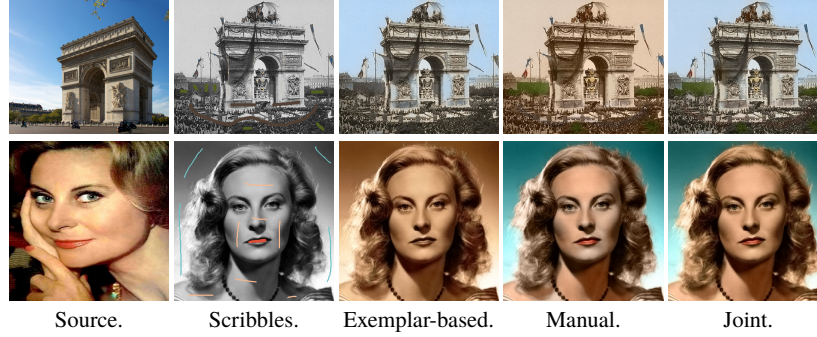
| Source. | Scribbles. | Exemplar-based. | Manual. | Joint. |

Fig. 11 Advantage of the joint approach, compared to manual and exemplar-based colorization. From left to right: source, target with scribbles added by the user, exemplar-based result, scribble-based results, and finally the joint approach.

5.2.1 Coupling the CNN with a Variational Method

In image colorization, convolutional neural networks can be used to compute in each pixel a set of possible colors and their associated probabilities [83]. However, since the final choice is made without taking into account the regularity of the image, this leads to halo effects. To improve this, we first propose to adapt the functional of [60] to the regularization of such results within the framework of colorization. The method of [60] being able to choose between several color candidates in each pixel, it will be quite easy to use on the color distribution provided by the CNN described in [83]. In addition, the numerical results of [60] demonstrate the ability to remove halos, which is relevant to the limitations of [83]. This functional will have to face two main problems: on the one hand, the transition from a low to a high resolution, and on the other hand, the maintenance of a higher saturation than current methods.

In this section, a method to couple the prediction power of CNN with the precision of variational methods is described. To this aim, let us remark that the variable w of the functional (27) represents the ratio of each color candidate which is represented in the final result. This comes from the fact that, for a given vector $w \in \mathbb{R}^C$, the minimum of

$$\sum_{i=1}^{C} w_i \|u - c_i\| \tag{33}$$

with respect to u is given by

$$\sum_{i=1}^{C} w_i c_i. \tag{34}$$

Thus, it can be seen as a probability distribution of the colors in the desired color image, which is exactly the same purpose as the one of the CNN in [83].

Figure 12 shows an overview of [54]. First, the gray-scale image, considered as the luminance L is given as an input to the CNN. The output of the CNN is a probability distribution over 313 possible chromaticity at low resolution (64×64). In
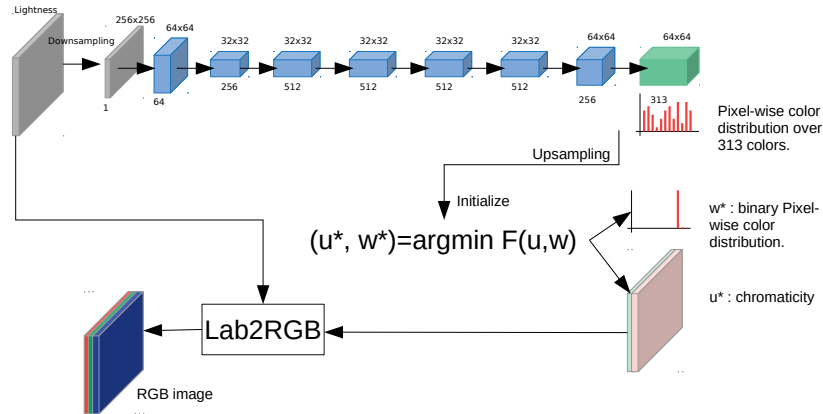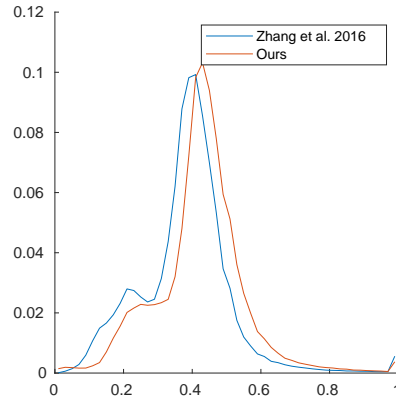
Fig. 12 Overview of method [54]. A CNN computes color distribution on each pixel. A variational method selects then a color for each pixel based on a regularity hypothesis.

order to initialize the minimization algorithm, the output weights of the CNN can be used. The CNN provides a coarse scale output, that needs an up-sampling before producing a suitable output at original definition. Two ways can be considered. For the first one, the variational method can be used at coarse scale (low definition), and then an interpolation can be performed to recover a result at fine scale (high definition). For the second one, the probability distributions can be interpolated to get a high definition array. In the following, the second approach will be preferred. Indeed, the interpolation of a color image produces a decrease of the saturation, that makes images drabber. By interpolating the probability distributions instead of the color images, the variational method will be able to compute a color for each pixel based on a coupling of the channels at high resolution. The given probability distribution is then used as initialization value for the numerical scheme. As it was still proposed in [60], the variable u is initialized with $\sum_{i=1}^{C} w_i c_i$. After the iterations of the functional, the result, denoted by (u^*, w^*), provides some binary weights (see, *eg*, [60], Section 2.3.2) and a regularized result u^* that gives two chromaticity channels, a and b, at initial definition. Recombined with the luminance L and transformed into the RGB space, that produces a color image.

Let us remark that the authors of [83] proposed to first produce the color image and then to resize it with bi-cubic interpolation. Unfortunately, up-sampling or down-sampling images with bi-linear or bi-cubic interpolations reduce the saturation of the colors and make them drabber than the original. To avoid that, we propose here the opposite approach: we first up-sample the color distribution, and then we compute a color image at full definition by using it. Since the numerical scheme is used at full definition, the required memory of the algorithm for all the weights and the colors is a limitation to process high resolution images on a standard PC. To tackle this issue,

| Original image | Zhang *et al.* 2016 | Mouzon *et al.* 2019 |

Histograms of saturation

Fig. 13 Results of Zhang *et al.* [83] compared with Mouzon *et al.* [54]. The histogram of the saturation shows the second result is shinier than the first one. Indeed, the average value of the saturation is higher for the model of Mouzon *et al.* [54] (0.4228) than the one of [83] (0.3802).

we propose to select some of the 313 colors. This selection is done with respect to the probability distribution of the colors, by choosing the 10 highest modes.

This choice of 10 modes has been done experimentally. For most images, 8 or 9 candidates are enough and taking more of them does not improve the result, but it increases the computational time. On the other hand, taking less candidates decreases the quality of the result on a significant number of images. Finally, the number of 10 is a fair trade-off.

The training step of the CNN is done as in [83]. The variational step is not taken into account during the training process. Indeed, the relation between the initialization of the weights and the result is not analytically described and the gradient back-propagation algorithms is not suitable for this problem. Thus, the training is done by feeding the CNN with a gray-scale image as input and a color distribution as output. The variational step remains independent of the full framework during the training step. Its integration will be the purpose of future works.

In the next section, numerical results are presented.

5.2.2 Numerical Results

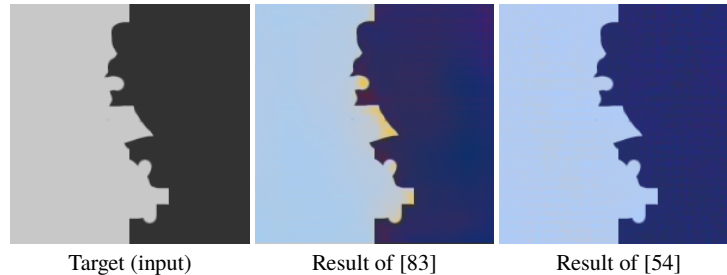Target (input)   Result of [83]   Result of [54]

Fig. 14 Comparison of [54] with [83]. This example provides a proof of concept. The method of [54] is able to remove the halo effects on the colorization result of [83].

In this section we show a qualitative comparison between [83] and the framework of [54]. A lot of results provided by [83] are accurate and reliable. We show on these examples that the method of [54] does not reduce the quality of the images. We then propose some comparisons with erroneous results of [83], which shows that the method of [54] is reliable to fully automatically colorize images without artifacts and halo effects. A time comparison between the CNN inference computation and the variational step will be proposed to show that the regularization of the result is not a burden on the CNN approach. Finally, to show the limitation of CNN in image colorization, we show some results where neither the approach of [83] nor the framework of [54] are able to produce some reliable results.

Figure 13 shows the colorization results of the method of Zhang *et al.* [83]. Whereas it is hard to see that the method of [54] produces a shinier result than the result of [83] unless being a calibration expert, the histogram of the saturation is able to show the improvement. Indeed, since the histogram is right-shifted, it means that globally, the saturation is higher on the result of [54]. Quantitatively, the average of the saturation is equal to 0.4228 for the method of [54], while it is equal to 0.3802 for the method of [83]. This improvement comes from the fact that the method of [54] selects one color among the ones given by the results of the CNN, whereas the method of [83] computes the annealed mean of them. The mean of the chrominances of the colors produces a decrease of the saturation and makes the colors drabber. By using a selection algorithm based on the image regularization, the method of [54] is able to avoid this drawback.

The result in Figure 14 is a proof of concept for the proposed framework. We can see a toy example which is automatically colorized by the method of [83]. The result given by the method of [83] produces some halo effect near the only contour of the image, which is unnatural. The regularization of the result is able to remove this halo effect and to recover an image looking less artificial. This toy example contains only two constant parts. The aim of the variational method is to couple the contours of
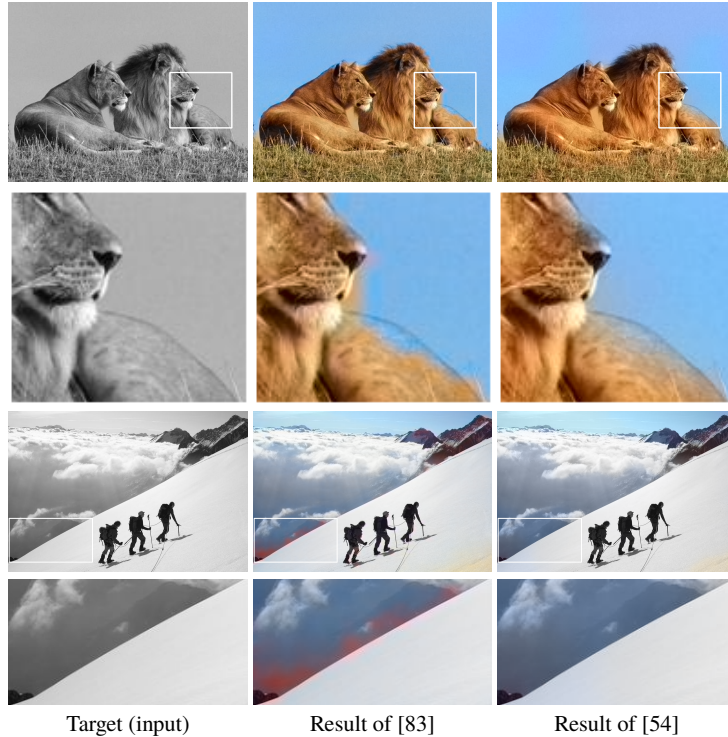
| Target (input) | Result of [83] | Result of [54] |

Fig. 15 Comparison between [54] and [83].

the chrominance channels and the ones of the luminance. The result produced with the method of [54] contains no halo effect, showing the benefits of their framework.

In Figure 15, we review some results and we compare them to the method of [83]. For the lion, (first line), a misalignment of the colors with the gray scale image is visible (a part of the lion is colorized in blue and a part of the sky is brown beige). This is a typical case of halo effect where the framework of [54] is able to remove the artifacts. For the image of mountaineer, on the result of [83] some pink stains appear. With the method of [54], the minimization of the total variation ensures the regularity of the image, thus it removes these strains.

Figure 16 shows additional results. The first line is an old port-card. Its colorization is reliable with the CNN and, in addition, the variational approach makes it a little bit shinier. This example shows the ability of the approach of [54] to colorize historical images. In the second example, most of the image is well colorized by the original method of [83]. Nevertheless, the lighthouse as well as the right-side building contain some orange halos that are not reliable. With the variational method, the colors are convincing. Additional results are available on `http://www.fabienpierre.fr/colorization`
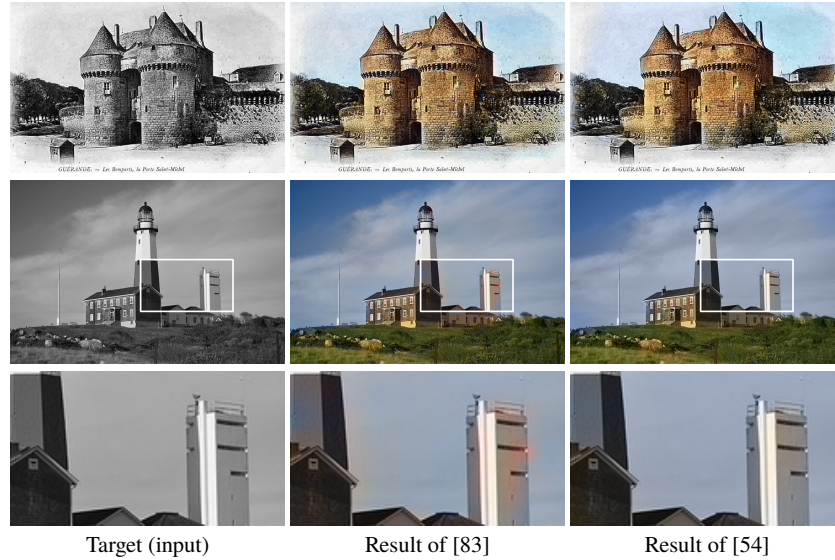
| Target (input) | Result of [83] | Result of [54] |

Fig. 16 Additional comparisons of [54] with [83].

The computational time of the CNN forward pass is about 1.5 sec in GPU, whereas the minimization of the variational model (27) is about 15 sec in Matlab on CPU. In [66], the authors provide a computation time almost equal to 1 sec with unoptimized GPU implementation. Since the minimization scheme of [77] is approximately the same, the computational time would be almost equal. Thus, the computational time of the approach of [54] is not a burden in comparison with the method of [83].

In Figure 17, a failure case is shown. In this case, since the minimization of the variational model strongly depends on its initialization, the method of [54] is not able to recover realistic colors. Actually, fully automatic colorization remains an open problem.

6 Conclusion and Future Works

In this chapter, we have shown that image colorization has known a huge progress during the last ten years by introducing a wide number of methods and approaches. Some extensions of these techniques have been proposed for video colorization but with limited number of frames. Future works could consider this application with more success. In this work, we have shown some limitations to colorization which let the topic open for active research. Joint approaches have shown their efficiency and a combination of deep leaning with manual approaches could enhance the human system interface for image and video colorization.
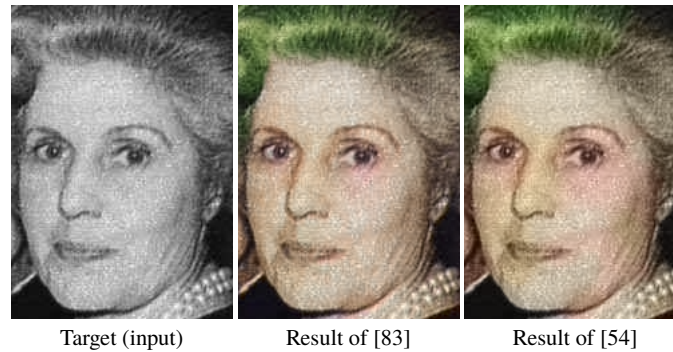
| Target (input) | Result of [83] | Result of [54] |

Fig. 17 Fail case. The prediction of the CNN is not able to recover a reliable color.

Acknowledgment

References

1. Abidi, B.R., Zheng, Y., Gribok, A.V., Abidi, M.A.: Improving weapon detection in single energy x-ray images through pseudocoloring. IEEE Transactions on Systems, Man, and Cybernetics, Part C **36**(6), 784–796 (2006)
2. Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., Süsstrunk, S.: Slic superpixels compared to state-of-the-art superpixel methods. IEEE Transactions on Pattern Analysis and Machine Intelligence **34**(11), 2274–2282 (2012)
3. Bay, H., Tuytelaars, T., Van Gool, L.: Surf: Speeded up robust features. In: European Conference on Computer Vision, pp. 404–417. Springer (2006)
4. Berkels, B., Effland, A., Rumpf, M.: Time discrete geodesic paths in the space of images. SIAM Journal on Imaging Sciences **8**(3), 1457–1488 (2015)
5. Bugeau, A., Ta, V.T.: Patch-based image colorization. In: IEEE International Conference on Pattern Recognition, pp. 3058–3061 (2012)
6. Bugeau, A., Ta, V.T., Papadakis, N.: Variational exemplar-based image colorization. IEEE Transactions on Image Processing **23**(1), 298–307 (2014)
7. Cao, Y., Zhou, Z., Zhang, W., Yu, Y.: Unsupervised diverse colorization via generative adversarial networks. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, pp. 151–166. Springer (2017)
8. Caselles, V., Facciolo, G., Meinhardt, E.: Anisotropic cheeger sets and applications. SIAM Journal on Imaging Sciences **2**(4), 1211–1254 (2009)
9. 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)
10. Chan, T.F., Kang, S.H., Shen, J.: Total variation denoising and enhancement of color images based on the cb and hsv color models. Journal of Visual Communication and Image Representation **12**(4), 422–435 (2001)
11. Chan, T.F., Vese, L.A.: Active contours without edges. IEEE Transactions on Image Processing **10**(2), 266–277 (2001)

12. Charpiat, G., Hofmann, M., Schölkopf, B.: Automatic image colorization via multimodal predictions. In: European Conference on Computer Vision, pp. 126–139. Springer (2008)
13. Chen, T., Wang, Y., Schillings, V., Meinel, C.: Grayscale image matting and colorization. In: Asian Conference on Computer Vision, pp. 1164–1169 (2004)
14. Chen, Y., Luo, Y., Ding, Y., Yu, B.: Automatic colorization of images from chinese black and white films based on cnn. In: 2018 IEEE International Conference on Audio, Language and Image Processing, pp. 97–102 (2018)
15. Chen, Y., Ye, X.: Projection onto a simplex. arXiv preprint arXiv:1101.6081 (2011)
16. Chia, A.Y.S., Zhuo, S., Kumar, R.G., Tai, Y.W., Cho, S.Y., Tan, P., Lin, S.: Semantic colorization with internet images. In: ACM SIGGRAPH ASIA (2011)
17. Comaniciu, D., Meer, P.: Mean shift: A robust approach toward feature space analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence **24**(5), 603–619 (2002)
18. Cui, M., Hu, J., Razdan, A., Wonka, P.: Color-to-gray conversion using isomap. The Visual Computer **26**(11), 1349–1360 (2010)
19. Deledalle, C.A., Papadakis, N., Salmon, J., Vaiter, S.: Clear: Covariant least-square re-fitting with applications to image restoration. SIAM Journal on Imaging Sciences (2017)
20. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 248–255 (2009)
21. Deshpande, A., Lu, J., Yeh, M.C., Chong, M.J., Forsyth, D.A.: Learning diverse image colorization. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 2877–2885 (2017)
22. Di Blasi, G., Reforgiato, D.: Fast colorization of gray images. Eurographics Italian (2003)
23. Ding, X., Xu, Y., Deng, L., Yang, X.: Colorization using quaternion algebra with automatic scribble generation. In: Advances in Multimedia Modeling (2012)
24. Drew, M.S., Finlayson, G.D.: Improvement of colorization realism via the structure tensor. International Journal of Image and Graphics **11**(04), 589–609 (2011)
25. 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)
26. Fitschen, J.H., Nikolova, M., Pierre, F., Steidl, G.: A variational model for color assignment. In: Scale Space and Variational Methods in Computer Vision, pp. 437–448 (2015)
27. Fornasier, M.: Nonlinear projection recovery in digital inpainting for color image restoration. Journal of Mathematical Imaging and Vision **24**(3), 359–373 (2006)
28. Gonzalez, R.C., Woods, R.E.: Digital image processing 3rd edition (2008)
29. Guadarrama, S., Dahl, R., Bieber, D., Shlens, J., Norouzi, M., Murphy, K.: Pixcolor: Pixel recursive colorization. In: British Machine Vision Conference (2017)
30. 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)
31. He, M., Chen, D., Liao, J., Sander, P.V., Yuan, L.: Deep exemplar-based colorization. ACM Transactions on Graphics **37**(4), 47:1–47:16 (2018)
32. 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)
33. Heu, J.H., Hyun, D.Y., Kim, C.S., Lee, S.U.: Image and video colorization based on prioritized source propagation. In: IEEE International Conference on Image Processing, pp. 465–468 (2009)
34. Iizuka, S., Simo-Serra, E., Ishikawa, H.: Let there be Color!: Joint End-to-end Learning of Global and Local Image Priors for Automatic Image Colorization with Simultaneous Classification. ACM Transactions on Graphics **35**(4) (2016)
35. Irony, R., Cohen-Or, D., Lischinski, D.: Colorization by example. In: Eurographics Symp. on Rendering, vol. 2. Citeseer (2005)
36. Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. In: IEEE Conference on Computer Vision and Pattern Recognition (2017)
37. Jin, Z., Min, L., Ng, M.K., Zheng, M.: Image colorization by fusion of color transfers based on dft and variance features. Computers & Mathematics with Applications (2019)

38. Jin, Z., Zhou, C., Ng, M.K.: A coupled total variation model with curvature driven for image colorization. Inverse Problems and Imaging **10**(1930-8337), 1037 (2016). DOI 10.3934/ipi.2016031

39. Jung, M., Kang, M.: Variational image colorization models using higher-order mumford–shah regularizers. Journal of Scientific Computing **68**(2), 864–888 (2016). DOI 10.1007/s10915-015-0162-9. URL https://doi.org/10.1007/s10915-015-0162-9

40. Kang, S.H., March, R.: Variational models for image colorization via chromaticity and brightness decomposition. IEEE Transactions on Image Processing **16**(9), 2251–2261 (2007)

41. Kawulok, M., Kawulok, J., Smolka, B.: Discriminative textural features for image and video colorization. IEICE Transaction on Information and Systems **95-D**(7), 1722–1730 (2012)

42. Kim, T.H., Lee, K.M., Lee, S.U.: Edge-preserving colorization using data-driven random walks with restart. In: IEEE International Conference on Image Processing, pp. 1661–1664 (2010)

43. Kuhn, G.R., Oliveira, M.M., Fernandes, L.A.: An improved contrast enhancing approach for color-to-grayscale mappings. The Visual Computer **24**(7-9), 505–514 (2008)

44. Kuzovkin, D., Chamaret, C., Pouli, T.: Descriptor-based image colorization and regularization. In: Computational Color Imaging, pp. 59–68 (2015)

45. Lagodzinski, P., Smolka, B.: Digital image colorization based on probabilistic distance transformation. In: 50th International Symposium ELMAR, vol. 2, pp. 495–498 (2008)

46. Lannaud, C.: Fallait-il coloriser la guerre ? L'express (2009). Disponible en ligne sur http://www.lexpress.fr/culture/tele/fallait-il-coloriser-la-guerre_789380.html

47. Larsson, G., Maire, M., Shakhnarovich, G.: Learning representations for automatic colorization. In: European Conference on Computer Vision, pp. 1–16. Springer (2016)

48. Levin, A., Lischinski, D., Weiss, Y.: Colorization using optimization. In: ACM Transactions on Graphics, vol. 23–3, pp. 689–694 (2004)

49. Lézoray, O., Elmoataz, A., Bougleux, S.: Graph regularization for color image processing. Computer Vision and Image Understanding **107**(1), 38–55 (2007)

50. Lézoray, O., Elmoataz, A., Meurie, C.: Mathematical morphology in any color space. In: IAPR/IEEE International Conference on Image Analysis and Processing, Computational Color Imaging Workshop (2007)

51. Lézoray, O., Meurie, C., Elmoataz, A.: A graph approach to color mathematical morphology. In: IEEE International Symposium on Signal Processing and Information Technology, pp. 856–861 (2005)

52. Lézoray, O., Ta, V.T., Elmoataz, A.: Nonlocal graph regularization for image colorization. In: IEEE International Conference on Pattern Recognition, pp. 1–4 (2008)

53. Luan, Q., Wen, F., Cohen-Or, D., Liang, L., Xu, Y.Q., Shum, H.Y.: Natural image colorization. In: Proceedings of the 18th Eurographics Conference on Rendering Techniques, EGSR'07, pp. 309–320. Eurographics Association, Aire-la-Ville, Switzerland, Switzerland (2007). DOI 10.2312/EGWR/EGSR07/309-320. URL http://dx.doi.org/10.2312/EGWR/EGSR07/309-320

54. Mouzon, T., Pierre, F., Berger, M.O.: Joint CNN and Variational Model for Fully-automatic Image Colorization. In: SSVM 2019 - Seventh International Conference on Scale Space and Variational Methods in Computer Vision. Hofgeismar, Germany (2019). URL https://hal.archives-ouvertes.fr/hal-02059820

55. Nikolova, M., Steidl, G.: Fast hue and range preserving histogram specification: Theory and new algorithms for color image enhancement. IEEE Transactions on Image Processing **23**(9), 4087–4100 (2014)

56. Perona, P., Malik, J.: Scale-space and edge detection using anisotropic diffusion. IEEE Transactions on Pattern Analysis and Machine Intelligence **12**(7), 629–639 (1990)

57. Persch, J., Pierre, F., Steidl, G.: Exemplar-based face colorization using image morphing. Journal of Imaging **3**(4), 48 (2017)

58. Peter, P., Kaufhold, L., Weickert, J.: Turning diffusion-based image colorization into efficient color compression. IEEE Transactions on Image Processing **26**(2), 860–869 (2017)

59. Peyré, G.: Toolbox fast marching - a toolbox for fast marching and level sets computations (2008). URL http://www.mathworks.com/matlabcentral/fileexchange/loadFile.do ?objectId=6110&objectType=FILE

60. Pierre, F., Aujol, J.F., Bugeau, A., Papadakis, N., Ta, V.T.: Luminance-chrominance model for image colorization. SIAM Journal on Imaging Sciences **8**(1), 536–563 (2015)
61. Pierre, F., Aujol, J.F., Bugeau, A., Ta, V.T.: Hue constrained image colorization in the RGB space. Preprint. (2014). Disponible en ligne sur https://hal.archives-ouvertes.fr/hal-00995724/document
62. Pierre, F., Aujol, J.F., Bugeau, A., Ta, V.T.: A unified model for image colorization. In: Color and Photometry in Computer Vision (ECCV Workshop), pp. 1–12 (2014)
63. Pierre, F., Aujol, J.F., Bugeau, A., Ta, V.T.: Combinaison linéaire optimale de métriques pour la colorisation d'images. In: XXVème colloque GRETSI, pp. 1–4 (2015)
64. Pierre, F., Aujol, J.F., Bugeau, A., Ta, V.T.: Luminance-hue specification in the RGB space. In: Scale Space and Variational Methods in Computer Vision, pp. 413–424 (2015)
65. Pierre, F., Aujol, J.F., Bugeau, A., Ta, V.T.: Colociel. Dépôt Agence de Protection des Programmes No IDDN.FR.001.080021.000.S.P.2016.000.2100 (2016). Disponible en ligne sur http://www.labri.fr/perso/fpierre/colociel_v1.zip
66. Pierre, F., Aujol, J.F., Bugeau, A., Ta, V.T.: Interactive video colorization within a variational framework. SIAM Journal on Imaging Sciences **10**(4), 2293–2325 (2017)
67. Pierre, F., Aujol, J.F., Bugeau, A., Ta, V.T., Papadakis, N.: Exemplar-based colorization in RGB color space. In: IEEE International Conference on Image Processing, pp. 1–5 (2014)
68. Pierre, F., Aujol, J.F., Deledalle, C.A., Papadakis, N.: Luminance-guided chrominance denoising with debiased coupled total variation. In: International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition, pp. 235–248. Springer (2017)
69. 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)
70. Ren, X., Malik, J.: Learning a classification model for segmentation. In: IEEE International Conference on Computer Vision, pp. 10–17 (2003)
71. Royer, A., Kolesnikov, A., Lampert, C.H.: Probabilistic image colorization. In: British Machine Vision Conference (2017)
72. Sethian, J.A.: Level set methods and fast marching methods: evolving interfaces in computational geometry, fluid mechanics, computer vision, and materials science, vol. 3. Cambridge university press (1999)
73. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: International Conference on Learning Representations (2015)
74. Song, M., Tao, D., Chen, C., Bu, J., Yang, Y.: Color-to-gray based on chance of happening preservation. Neurocomputing **119**, 222–231 (2013)
75. Su, Z., Liang, X., Guo, J., Gao, C., Luo, X.: An edge-refined vectorized deep colorization model for grayscale-to-color images. Neurocomputing (2018)
76. Sỳkora, D., Buriánek, J., Žára, J.: Unsupervised colorization of black-and-white cartoons. In: Proceedings of the 3rd international symposium on Non-photorealistic animation and rendering, pp. 121–127. ACM (2004)
77. Tan, P., Pierre, F., Nikolova, M.: Inertial alternating generalized forward–backward splitting for image colorization. Journal of Mathematical Imaging and Vision pp. 1–19 (2019)
78. 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)
79. Welsh, T., Ashikhmin, M., Mueller, K.: Transferring color to greyscale images. In: ACM Transactions on Graphics, vol. 21–3, pp. 277–280. ACM (2002)
80. Williams, A., Barrus, S., Morley, R.K., Shirley, P.: An efficient and robust ray-box intersection algorithm. In: ACM SIGGRAPH 2005 Courses, p. 9 (2005)
81. Wolfgang Baatz Massimo Fornasier, P.A.M., , Schönlieb, C.B.: Inpainting of ancient austrian frescoes. In: R. Sarhangi, C.H. Séquin (eds.) Bridges Leeuwarden: Mathematics, Music, Art, Architecture, Culture, pp. 163–170. Tarquin Publications, London (2008). Disponible en ligne sur http://archive.bridgesmathart.org/2008/bridges2008-163.html
82. Yatziv, L., Sapiro, G.: Fast image and video colorization using chrominance blending. IEEE Transactions on Image Processing **15**(5), 1120–1129 (2006)

83. Zhang, R., Isola, P., Efros, A.A.: Colorful image colorization. In: European Conference on Computer Vision, pp. 1–16. Springer (2016)
84. Zhang, R., Zhu, J.Y., Isola, P., Geng, X., Lin, A.S., Yu, T., Efros, A.A.: Real-time user-guided image colorization with learned deep priors. ACM Transactions on Graphics **9**(4) (2017)
85. Zheng, Y., Essock, E.A.: A local-coloring method for night-vision colorization utilizing image analysis and fusion. Information Fusion **9**(2), 186–199 (2008)