Interactive Video Colorization within a Variational Framework

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

INTERACTIVE VIDEO COLORIZATION WITHIN A VARIATIONAL FRAMEWORK

F. PIERRE[†‡§¶], J.-F. AUJOL[†‡], A. BUGEAU[§¶], AND V.-T. TA[¶‖]

Abstract. This paper deals with the difficult problem of video colorization. Methods in the literature are generally based on spatio-temporal video blocks, or on frame-to-frame color propagation methods, each technique having its own advantages and drawbacks. In this paper, we present both a novel automatic frame-to-frame propagation approach and an interactive correction method within a variational framework. The proposed method propagates colors from an initial colorized frame to the whole grayscale video sequence. The automatic propagation results may be visually unsuitable in some cases. To overcome this limitation, a spatio-temporal functional with a user-guided correction is introduced. Two fast primal-dual algorithms are designed to solve the proposed variational models. Numerical results show the efficiency and the potentiality of the proposed approach in comparison with state-of-the-art methods.

Key words. colorization, optimization, non-local methods

AMS subject classifications. 68U10, 94A08, 65K10, 49M29

1. Introduction.

To restore old black-and-white movies and make them more attractive, among young people for instance, cinema and entertainment industries frequently broadcast colorized versions. In France, in 2014, *Apocalypse*, a historical documentary by I. Clarke and D. Costelle was realized from archives colorized by F. Montpellier. The broadcast gathered over 18.5% of viewers over the age bracket 11-14 during the first two episodes [21]. 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.

In this work, we assume that a grayscale video is available and one of its frames is colorized. This frame can be colorized by an expert or automatically [37, 28, 29]. Video colorization results have to be visually natural on both constant and textured parts, while a temporal consistency has to be respected.

Traditionally, colorization methods (image or video) assume that the grayscale image corresponds to a luminance channel Y. The Y channel is defined as a weighted average of the RGB channels: $Y = 0.299R + 0.587G + 0.114B$. The luminance-chrominance spaces propose to integrate the Y channel with two other ones, called chrominances. In practice, given the luminance channel Y, colorization methods estimate two chrominance channels before converting into the RGB space while keeping the luminance Y unchanged. Luminance-chrominance spaces usually used in image or video colorization are the $l\alpha\beta$ [9], YUV [22] or YCbCr [38].

For image colorization, state-of-the-art methods can be divided into three categories: the first ones that diffuse scribbles over the image (*e.g.*, [22, 38]), the second ones that use non-local techniques (*e.g.*, [37, 15]), and the others combining the two approaches (*e.g.*, [18, 28]).

Extension from image to video colorization are also proposed. State-of-the-art methods can be divided into two categories: the ones that diffuse colors from scribbles

[†]Univ. Bordeaux, IMB, UMR 5251, F-33400 Talence, France.
[‡]CNRS, IMB, UMR 5251, F-33400 Talence, France.
[§]Univ. Bordeaux, LaBRI, UMR 5800, F-33400 Talence, France.
[¶]CNRS, LaBRI, UMR 5800, F-33400 Talence, France.
[‖]Bordeaux INP, LaBRI, UMR 5800, F-33402 Talence, France.

over a three dimensional block (2D images + time), and the others that propagate colors from one frame to its adjacent ones until the whole video sequence is colorized. This last category is called *frame-to-frame propagation* in this paper.

The three dimensional diffusion approaches.. The colorization methods described in this paragraph perform results with a visually natural temporal consistency.

To recover a colorized video, it is required to compute the chrominance channels. In the seminal paper [22], Levin *et al.* propose to propagate the values of chrominances of some scribbles given by a user. The propagation over the whole video sequence is performed by minimizing a quadratic function that favours a coupling of luminance and chrominance contours. This criterion is based on neighbor pixels differences in both spatial and temporal dimensions. Neighborhoods in the temporal dimension are built after image registration from Lucas-Kanade optical flow estimation [23]. By considering three dimensional blocks, the method naturally deals with occlusions and dis-occlusions. Inspired from [18], Zhen *et al.* [39] extend the method of [22] to automatic exemplar-based (see, *e.g.*, [37]) video colorization. The approach of [22] being dependent on the optical flow computation, Lang *et al.* [20] propose to compute a more robust estimation, based on an energy minimization, to improve colorization results.

The initial quadratic function of [22] is not adapted to textured images. Sheng *et al.* [33] replace the spatial distance by a function depending on Gabor features [24]. The definition of the optical flow is also extended to the Gabor feature space. The method can deal with textures and is more robust to the noise than the original function of [22].

All these techniques are based on the minimization of a quadratic criterion computed on the whole video sequence.

Heu *et al.* [16] diffuse the chrominances of the scribbles to the other pixels with a priority order. The method estimates the reliability of a color for each pixel to its neighbors. For the video colorization, the reliability is computed between frames with a block-matching approach. Hyun *et al.* [17] extend [16] and modify the reliability within a multi-scale framework. These two last methods perform suitable results on smooth images but the extension to textured videos is only proposed as a perspective.

Finally, two methods interpolate chrominance channels. Yatziv *et al.* [38] blend the colors of the scribbles according to the spatio-temporal geodesic distance from one pixel to each scribble.

For textured images, Kang *et al.* [30] use the Reproducing Kernel Hilbert Space for the interpolation of chrominances on textured images. The video colorization is performed by extending the functions to the three dimensions.

Methods working on the three dimensional blocks can deal with occlusion and dis-occlusion problems. In contrast, the interactivity is difficult to reach due to the large amount of pixels to process and the computational burden. Indeed, the whole video sequence has to be processed and checked by the user after scribbling. The division of the video in some smaller sequences can be considered, but the concatenation of the colorized sequences may produce temporal inconsistencies.

To tackle these issues, the frame-to-frame approaches are presented in the following.

The frame-to-frame approaches.. In the seminal paper [37], Welsh *et al.* apply their exemplar-based image colorization method to each frame of a video sequence. This method is able to perform neither temporal nor spatial consistency.

Sykora *et al.* [34] address the problem of cartoon colorization. This approach

TABLE 1
Summary of the state-of-the-art methods. We propose to merge all their advantages into a variational framework.

Methods	Patch based	Optical Flow	Regularization
[34]	√	×	√
[19, 16, 17]	√	×	×
[33, 39, 20, 22]	×	√	√
[35]	×	√	×
[25, 38, 30]	×	×	√
Our	√	√	√

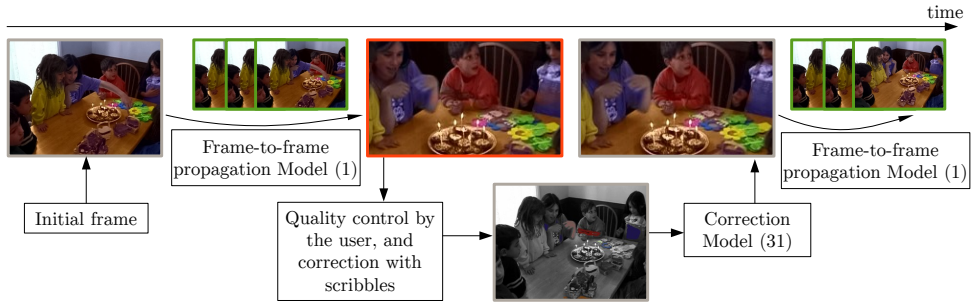


FIG. 1. *Overview of the whole approach. For most of the sequence, the frame-to-frame propagation model computes suitable results (in green). In the case of dis-occlusion area, a user-guided approach is mixed with the unsuitable result (in red). After that, the frame-to-frame propagation restarts from the visually suitable frame.*

is based on patch comparisons and manage large displacements and rotations. The correspondence map between patches not being dense, it is not adapted for textured images and complex motion of natural videos.

Jacob *et al.* [19] propose to colorize an image from scribbles given by a user. To propagate the color of a frame to the whole video sequence, a search in the previous frame for the closest patch is performed according to the sum of squared differences distance, and the color of the best matching pixel is considered. The search for the closest patch is computed in a neighborhood depending on a motion estimation.

In Pan *et al.* [25], the color is transferred to the adjacent frames, based on the motion estimation of [4]. Teng *et al.* [35] propose to use a refined block-matching algorithm to propagate colors over the frames. The user has to choose when the refined version is used instead of the original one.

All the frame-to-frame approaches use a specific motion estimation, each having its own advantage. These methods are sensitive to mistakes since the result is re-used and unsuitable colors are propagated.

In this paper, we propose a variational framework that merges the advantages of different correspondence maps including motion estimations and reduces the propagation of unsuitable colors.

Temporal consistency for video colorization.. State-of-the-art approaches generally use at least one of these techniques: patch correspondence, optical flow algorithm or regularization. Each of these techniques have their own advantages and drawbacks. The optical flow is reliable because it is based on physical interpretation of

the motion through the illumination equation. In contrast, it is not able to deal with large displacement or dis-occlusion. The computation of the neighbor map from patch comparison is able to tackle this issue, but being not based on physical assumption, it is more artificial than a real motion, which has to be continuous. The methods based on regularization are not adapted to textured or noisy sequences, but can deal with the occlusion and dis-occlusion problems. The comparisons between all the cited state-of-the-art approaches are summarized in Table 1.

Contributions.. In this paper, we first propose a frame-to-frame variational model and a primal-dual like algorithm to compute a solution. The method uses both color and map regularization to propagate colors. It is based on one or many correspondence maps. In this work, for the sake of clarity, we only focus on the case of two correspondence maps without loss of generality.

Next, we explain how the previous frame-to-frame propagation model can be extended to user interaction where color scribbles can be added on an unsuitable colorized frame of the video sequence. The user correction is merged with the previous result produced by the frame-to-frame propagation model. Contrary to the proposed propagation, the correction one is performed over a three dimensional representation of the video sequence to solve occlusion or dis-occlusion problems.

Outline.. In Section 2, we describe a variational approach computing a regularized result from the correspondence maps. This model is solved by a primal-dual like algorithm. A scribble correction technique is proposed in Section 3, where a primal-dual algorithm is designed to solve it. In Section 4, we describe some implementation details, in particular, we explain how the two proposed models interact with each other. Figure 1 summarizes the basis of our approach. Finally, in Section 5 we present results and comparisons with state-of-the-art methods.

2. Frame-to-Frame Propagation Model.

In this section we propose to introduce a new functional for video colorization that propagates colors from a frame to an adjacent one. The proposed method computes an optimal correspondence map between the video frames, based on a trade-off between the regularity of the map and the resulting color. This regularization is performed with total variation (TV).

2.1. Overview of the Frame-to-Frame Propagation Method .

Our approach combines correspondence maps that can be obtained with different techniques (*e.g.*, [3, 14]). Correspondence maps enable the mapping of chrominance values and the propagation of colors through the video sequence. Our approach considers multiple correspondence maps and the proposed model select the best one. As illustrated in Figure 2, the first step of our frame-to-frame propagation method computes the correspondence maps, which can be performed as a pre-processing. Each correspondence map provides a color candidate.

Next, from a colorized initial frame, the method estimates the best candidate by minimizing the functional described in Section 2.2. The minimization provides both the colorized frame and the optimal correspondence map. They are both regularized by a TV term. This aspect is detailed in Section 2.3.

Figure 2 summarizes our frame-to-frame propagation approach.

2.2. A New Functional Based on Color Regularization and Correspondence Maps.

In this section, a model performing both a choice between $\mathcal{M}$ correspondence maps and computing the color of a frame is proposed.

A video, denoted by u is assumed to be a finite sequence of T frames, indexed in time by a discrete variable $t \in [\![1; T]\!]$. The video is considered in RGB and converted

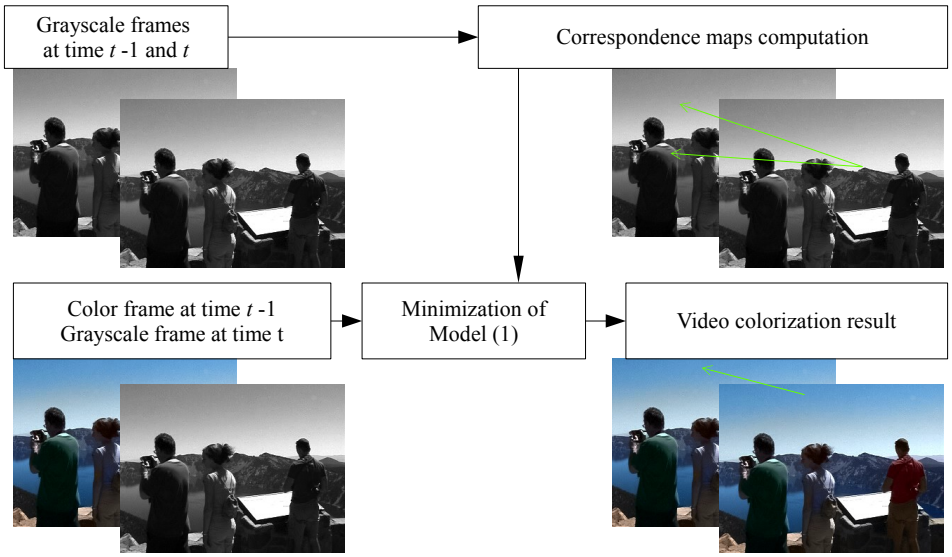


FIG. 2. *Overview of the frame-to-frame propagation method. In this example, two correspondence maps between grayscale frames at times $t-1$ and t are computed in a first step. The frame-to-frame propagation method estimates both an optimal correspondence map and the final color for the frame at time t. The green arrows represent the correspondences between two frames for a considered pixel. The first step provides two correspondence maps, while the minimization of Functional computes the optimal one.*

into the YUV color space. $u^{(t)}$ denotes the chrominance channels (U, V) of the frame at time t. The initial grayscale video sequence is supposed to be the luminance channel of the final colorized video. The computation of the chrominance channels U and V enables to recover the RGB corresponding colors. The image domain is denoted as Ω.

For each pair of frames $u^{(t-1)}$ and $u^{(t)}$, the proposed functional chooses, at each position x, one correspondence map between the $\mathcal{M}$ available.

This selection of the best correspondence is computed within a variational framework. The related work of Pierre *et al.* [27] performs a choice between different colors under an assumption of regularity of the final result. We extend this model for both correspondence maps and the final color, and we propose the following new functional:

$$(\hat{u}^{(t)}, \hat{w}^{(t)}) = \operatorname{argmin}_{u^{(t)}, w^{(t)}} \alpha\, \mathrm{TV}_{\mathfrak{C}}(u^{(t)}) + \beta\, \mathrm{TV}(v^{(t)}) + \lambda \int_{\Omega} \sum_{i=1}^{\mathcal{M}} w_i^{(t)}(x) \|u^{(t)}(x) - c_i^{(t)}(x)\|_2^2 \, dx + \chi_{\mathcal{R}}(u^{(t)}) + \chi_{\Delta}(w^{(t)}). \tag{1}$$

$c_i^{(t)}$ is the color candidate given from the frame $u^{(t-1)}$, by the correspondence map $v_i^{(t)}$, thanks to its definition available in Equation (5) and (6). A trade-off between the regularity of the map $v^{(t)}$ and the colors of the result $u^{(t)}$ is controlled by the parameters α, β and λ. $w^{(t)}$ is a weight parameter that measures the pixel-wise contribution between the correspondence maps.

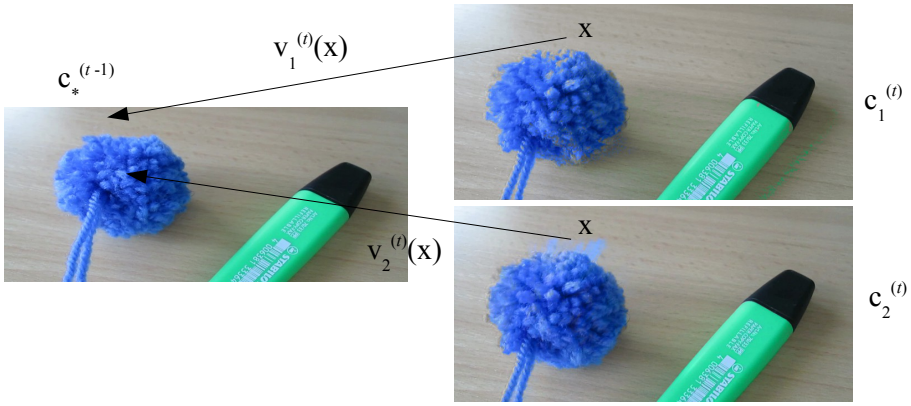


FIG. 3. *Example of correspondence maps between two frames ($\mathcal{M} = 2$). From a given result at time $t-1$, two possible propagation results are provided at time t from the two maps. The frame-to-frame model computes a regularized optimal final color.*

$\mathrm{TV}_{\mathfrak{C}}(u^{(t)})$ is a regularization term based on the minimization of the total variation [31] of chrominance channels of the frame $u^{(t)}$ at time t:

$$\mathrm{TV}_{\mathfrak{C}}(u^{(t)}) = \int_\Omega \left(\gamma\|\nabla Y\|_2^2 + \|\nabla U\|_2^2 + \|\nabla V\|_2^2\right)^{1/2}, \tag{2}$$

where $\nabla = (\partial_x, \partial_y)$ is the spatial gradient of a frame. The parameter γ controls the coupling of the luminance channel Y with the chrominance ones, avoiding halo effects [27].

$\mathrm{TV}(v^{(t)})$ enforces the regularity of the final correspondence map and is detailed in Section 2.3 (see Equation (7)). The correspondence map at time t denoted by $v^{(t)}$ is defined as a weighted average of the $v_i^{(t)}$, $i = 1, \ldots, \mathcal{M}$:

$$v^{(t)} = \sum_{i=1}^{\mathcal{M}} w_i^{(t)} v_i^{(t)}. \tag{3}$$

Finally, $\chi_{\mathcal{R}}(u^{(t)})$ guaranties that the minimizer of the functional is in the chrominance standard range (see, *e.g.*, [27]) and $\chi_\Delta(w^{(t)})$ constrains $w^{(t)}$ to be in the probability simplex (*i.e.*, $0 \leq w_i^{(t)} \leq 1$ and $\sum_{i=1}^{\mathcal{M}} w_i^{(t)} = 1$).

The definition of the data-fidelity term is based on the set of colors of the initial frame. Let us denote the propagated color $\hat{c}^{(t-1)}$ at time $t-1$ as:

$$\hat{c}^{(t-1)} = \sum_{i=1}^{\mathcal{M}} \hat{w}_i^{(t-1)} c_i^{(t-1)}. \tag{4}$$

Therefore, the candidates in the data-fidelity term are:

$$c_i^{(t)}(x) = \hat{c}^{(t-1)}(v_i^{(t)}(x)),\ i = 1, ..., \mathcal{M}, \tag{5}$$

and

$$c_i^{(1)}(x) = u^{(1)}(v_i^{(1)}(x)),\ i = 1, ..., \mathcal{M}. \tag{6}$$

With these definitions, Functional (1) computes an optimal map between the frames $u^{(t-1)}$ and $u^{(t)}$ with a regularization of both maps and colors.

Once the minimum of the functional with respect to $w^{(t)}$ is reached, the values of the optimum $\hat{w}^{(t)}$ are projected onto the canonical basis (*i.e.* using a a winner-takes-all approach), which avoids the melting of colors. In practice, the $w_i^{(t)}$ which has the maximum value gets 1, while the others get 0. Hence, Equations (6) and (5) enforce the data-fidelity term to be composed of colors coming from the initial frame. The definition of $\hat{c}^{(t-1)}$ preserves the set of colors of the initial frame, *i.e.*, the chrominances used at time t are only the initial ones. Indeed, since the weights are 0 or 1, by Equation (5), $\hat{c}^{(t)}$ is one of the color candidates of the previous frame. Thus, by induction, the data-fidelity term at time t is only composed with colors of the initial frame. This induction is illustrated in Figure 4. Since the regularization of the chrominance channels may produce new colors, the preservation of initial frame colors is important.

Figure 3 illustrates the computation of the data-fidelity term. While the $\mathcal{M}$ initial maps provide $\mathcal{M}$ colors, based on the previous result, the frame-to-frame model computes the optimal one.

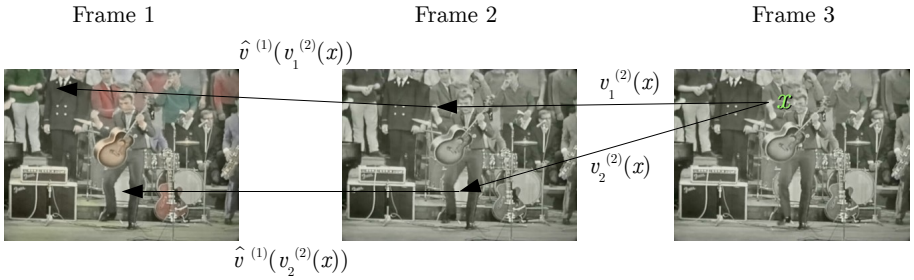


FIG. 4. *Illustration of the color propagation from the first frame. The colors are transferred from frame 1 to frame 2 with respect to the optimal map* $\hat{v}^{(1)} = \sum_{i=1}^{\mathcal{M}} \hat{w}_i^{(1)} v_i^{(1)}$ *computed by the proposed frame-to-frame propagation algorithm. The data-fidelity term available for a frame is only composed of chrominances existing in the previous one. By induction, the successive data-fidelity term are only composed of colors available in frame 1.*

With Functional (1), the computation of the solution only depends on the result of the optimization at the previous frame. The computations are performed iteratively from the initial frame to the last one. The minimization of the functional provides an optimal map among the initial ones (see Equation (3)).

The TV regularization of the correspondence maps is detailed in the next section.

2.3. Correspondence Maps Regularization. We propose to regularize both correspondence maps and colors of the result. Similar approach for map regularization is proposed by [13] to compute an aggregation of optical flows. In [13], the result is only based on the map regularization whereas our model considers both maps and chrominances regularity. In [11], a similar technique enables both the estimation of the optical flow and the denoising of the image. It differs from video colorization where the optical flow is computed on the luminance channel, whereas the final estimation is performed on chrominance channels. Other approaches preserving geometric structures have been proposed for inpainting problems [2, 1].

The correspondence map v is a two dimensional vector corresponding to the relative displacement of the objects between two frames. The two coordinates of v are denoted by φ (horizontal displacement) and ψ (vertical displacement).

The map regularization is based on the following term in Functional (1):

$$\mathrm{TV}(v) = \|\nabla v\|_{1,2} = \int_\Omega \|\nabla v(x)\|_2 \, dx. \tag{7}$$

The total variation of v favors constant parts of the correspondence maps. This regularization term provides a map preserving the textures from the initial frame. If the relative correspondence map is piece-wise constant between two frames, some parts of these frames have a similar content. Therefore, it mimics a copy-paste method from a frame to the next one. When this map is used to propagate the values of chrominance channels, some parts of adjacent frames that have similar luminance are colorized with the same chrominances. With this approach, the richness of the color in textures is preserved.

In the following, without loss of generality, and for the sake of clarity, we only consider the case of two correspondence maps ($\mathcal{M} = 2$). The generalization to multiple correspondence maps is straightforward. The computation of (3) can be simplified with the following parametrization:

$$w_1 = w,\ w_2 = 1 - w, \tag{8}$$

with $w \in [0, 1]$.

With Parametrization (8):

$$v = w(v_1 - v_2) + v_2. \tag{9}$$

To compute the total variation, the gradient is defined as:

$$\nabla v = (v_1 - v_2) \otimes \nabla w + w(\nabla v_1 - \nabla v_2) + \nabla v_2, \tag{10}$$

where $\otimes$ is the Kronecker product:

$$(v_1 - v_2) \otimes \nabla w = \begin{pmatrix} (\varphi_1 - \varphi_2)\partial_x w \\ (\varphi_1 - \varphi_2)\partial_y w \\ (\psi_1 - \psi_2)\partial_x w \\ (\psi_1 - \psi_2)\partial_y w \end{pmatrix}, \tag{11}$$

and $v_i = (\varphi_i, \psi_i)^T$.

We denote the linear operator on the variable w, in Equation (11), by A. Pixel-wise:

$$Aw = \underbrace{\begin{pmatrix} \varphi_1 - \varphi_2 & 0 \\ 0 & \varphi_1 - \varphi_2 \\ \psi_1 - \psi_2 & 0 \\ 0 & \psi_1 - \psi_2 \end{pmatrix}}_{:= A_1} \nabla w + \underbrace{\begin{pmatrix} \partial_x\varphi_1 - \partial_x\varphi_2 \\ \partial_y\varphi_1 - \partial_y\varphi_2 \\ \partial_x\psi_1 - \partial_x\psi_2 \\ \partial_y\psi_1 - \partial_y\psi_2 \end{pmatrix}}_{:= A_2} w. \tag{12}$$

With these notations, $\mathrm{TV}(v)$ reads as $\|Aw + \nabla v_2\|_{1,2}$.

The following lemma provides the operator norm of A, which is required for the implementation of the minimization algorithm.

LEMMA 2.1. *The operator norm of A is bounded by*

$$\|A\| \leq 2\sqrt{8}(N+M) + 4(N+M), \tag{13}$$

where M and N are the dimensions of a frame.

The proof of this lemma is available in Appendix A.

2.4. Algorithm for Color Propagation. In this section, we propose an iterative scheme solving Functional (1) with $\mathcal{M} = 2$.

2.4.1. General algorithm. For a proper and lower semi-continuous (*lsc*) functional F on $\mathbb{R}^P$ with values in $\mathbb{R} \cup \{+\infty\}$ let us define F^*, the Legendre-Fenchel transform, as follows:

$$F^*(p) = \max_{u \in \mathbb{R}^U} \langle p|u\rangle - F(u). \tag{14}$$

To design an algorithm for Model (1), we have to solve problems of the form:

$$\min_{u \in \mathbb{R}^U} \max_{p \in \mathbb{R}^P} \langle Ku|p\rangle - G^*(p) + F(u), \tag{15}$$

where G^* and F are proper, *lsc* and convex functions, and K is a linear and continuous operator that maps $\mathbb{R}^U$ on $\mathbb{R}^P$. Chambolle and Pock [7] propose a primal-dual algorithm to solve this class of problems.

In this work, we solve a problem of the form:

$$\min_{u \in \mathbb{R}^U, w \in \mathbb{R}^W} \max_{p \in \mathbb{R}^P, z \in \mathbb{R}^Z} F(u) + \langle Ku|p\rangle - G^*(p) + h(u,w) + H(w) + \langle Aw|z\rangle - J^*(z) \tag{16}$$

with A linear and continuous operator, F, H, G and J are proper, *lsc* and convex functions, h is a proper, *lsc* function, convex with respect to each of its variables.

To compute a saddle-point of Model (16), we propose to use Algorithm 1. σ_u, σ_w,

Algorithm 1 Primal-dual algorithm solving (16).

for $n \geq 0$ **do**
  $p^{n+1} \leftarrow \text{prox}_{\sigma_u G^*}\left(p^n + \sigma_u K \overline{u}^n\right)$
  $z^{n+1} \leftarrow \text{prox}_{\sigma_w J^*}\left(z^n + \sigma_w A \overline{w}^n\right)$
  $w^{n+1} \leftarrow \text{prox}_{\tau_w (H + h(u^n, .))}\left(w^n - \tau_w A^* z^{n+1}\right)$
  $u^{n+1} \leftarrow \text{prox}_{\tau_u (F + h(., w^n))}\left(u^n - \tau_u K^* p^{n+1}\right)$
  $\overline{u}^{n+1} \leftarrow 2u^{n+1} - u^n$
  $\overline{w}^{n+1} \leftarrow 2w^{n+1} - w^n$
end for

τ_w and τ_u are time steps. The proximal operator of a proper, *lsc* and convex function $f : \mathbb{R}^U \to \mathbb{R}$ is defined as (see, *e.g.* [10]):

$$\text{prox}_f(v) = \text{argmin}_{u \in \mathbb{R}^U} \|u - v\|_2^2 + f(u). \tag{17}$$

2.4.2. Dual version of Model (1) . To solve Model (1), the $1,2-$norm, recalled in Equation (7), is written in the dual form:

$$\beta\|u\|_{1,2} = \max_{p\in\mathbb{R}^P} \langle u|p\rangle - \chi_{B_{\mathbb{R}^P}(0,\beta)}(p), \tag{18}$$

with $u \in \mathbb{R}^U$, and $B_{\mathbb{R}^P}(0,\beta)$ the β radius ball in $\mathbb{R}^P$ with L^2-norm.

Once again, for the sake of clarity we consider $\mathcal{M} = 2$ in the following, but the extension to higher values of $\mathcal{M}$ is straightforward. Let us rewrite the total variation of the correspondence map (see Equation (7)), where n is the number of pixel, the correspondence maps are identified to vectors of $\mathbb{R}^{n\times 2}$:

$$\begin{aligned}\mathrm{TV}(v) &= \|(v_1 - v_2)\otimes\nabla w + w\,(\nabla v_1 - \nabla v_2) + \nabla v_2\|_{1,2}\\ &= \max_{z\in\mathbb{R}^{N\times 4}} \langle (v_1 - v_2)\otimes\nabla w + w\,(\nabla v_1 - \nabla v_2) + \nabla v_2|z\rangle\\ &\quad - \chi_{B_{\mathbb{R}^P}(0,\beta)}(z)\\ &= \max_{z\in\mathbb{R}^{N\times 4}} \langle Aw|z\rangle + \langle \nabla v_2|z\rangle - \chi_{B_{\mathbb{R}^P}(0,\beta)}(z),\end{aligned} \tag{19}$$

with A defined in (12).

Let us compute the dual operator A^* of A as follows:

$$A^* = A_1^* + A_2^*, \tag{20}$$

with the pixel-wise multiplication:

$$(\nabla v_1 - \nabla v_2)^* = \nabla v_1 - \nabla v_2, \tag{21}$$

and with:

$$((v_1 - v_2)\otimes\nabla)^* = \mathrm{div}(I_2 \otimes (v_1 - v_2)^T), \tag{22}$$

where $(v_1 - v_2)^T$ is equal to the transpose of the matrix A_1 defined in (12), and I_2 is the identity matrix of size 2.

Model (1) is rewritten in the primal-dual form:

$$\begin{aligned}\min_{u^{(t)},w^{(t)}} \max_{p,z} &\langle p(x)|\nabla u\rangle + \langle Aw|z\rangle + \langle \nabla v_2|z\rangle\\ &+ \lambda\int_\Omega w\|u - c_1\|_2^2 + (1-w)\|u - c_2\|_2^2\\ &\quad - \chi_{B(0,\alpha)}(p) - \chi_{B_{\mathbb{R}^P}(0,\beta)}(z) + \chi_{\mathcal{R}}(u) + \chi_{[0,1]}(w).\end{aligned} \tag{23}$$

2.4.3. Final reading of the algorithm. Since the terms of Equation (23) are pixel-wise separable, we then remove the $N \times M$ notations. We apply Algorithm 1 to (23) with the following identifications:

- $F(u) = \chi_{\mathcal{R}}(u)$
- $G^*(p) = \chi_{B_{\mathbb{R}^6}(0,\alpha)}(p)$
- $H(w) = \chi_{[0,1]}(w)$
- $J^*(z) = \chi_{B_{\mathbb{R}^4}(0,\beta)}(z) - \langle \nabla v_2|z\rangle$
- $h(u,w) = \lambda\left(w\|u - c_1\|_2^2 + (1-w)\|u - c_2\|_2^2\right)$

The proximal operators are given by the following lemmas.

LEMMA 2.2. *The proximal operator of $\sigma_w J^*$ is:*

$$\text{prox}_{\sigma_w J^*} = P_{B_{\mathbb{R}^{N\times 4}}(0,\beta)}(\tilde{z} + \sigma_w \nabla v_2), \tag{24}$$

where $P_{B_{\mathbb{R}^{N\times 4}}(0,\beta)}$ is the pixel-wise projection onto the L^2 ball of radius β.

Proof. The function is $-\sigma_w \langle \nabla v_2 | . \rangle + \chi_{B_{\mathbb{R}^{N\times 4}}(0,\beta)}$.

$$\begin{aligned}\text{prox}_{-\sigma_w\langle \nabla v_2|.\rangle + \chi_{B_{\mathbb{R}^{N\times 4}}(0,\beta)}}(\tilde{z}) \\ = \text{argmin}_{z\in\mathbb{R}^4} \frac{\|z-\tilde{z}\|_2^2}{2\sigma_w} - \langle \nabla v_2 | z\rangle + \chi_{B_{\mathbb{R}^{N\times 4}}(0,\beta)}(z),\end{aligned} \tag{25}$$

thus:

$$\text{prox}_{-\sigma_w\langle \nabla v_2|.\rangle + \chi_{B_{\mathbb{R}^{N\times 4}}(0,\beta)}}(\tilde{z}) = P_{B_{\mathbb{R}^{N\times 4}}(0,\beta)}(\tilde{z} + \sigma_w \nabla v_2). \tag{26}$$ ∎

LEMMA 2.3. *The proximal operator of $\tau_w h(u,w) + \tau_w H(w)$ is pixel-wise:*

$$\text{prox}_{\tau_w(h(u,w)+H(w))}(\tilde{w}) = P_{[0,1]}\left(\tilde{w} - \tau_w\lambda(\|u-c_1\|_2^2 - \|u-c_2\|_2^2)\right), \tag{27}$$

where $P_{[0,1]}$ is the orthogonal projection onto $[0,1]$.

Proof. The function is $\lambda w\|u-c_1\|_2^2 + \tau_w\lambda(1-w)\|u-c_2\|_2^2 + \chi_{[0,1]}(w)$:

$$\begin{aligned}\text{prox}_{\tau_w(h(u,w)+H(w))}(\tilde{w}) &= \text{argmin}_w \frac{\|w-\tilde{w}\|_2^2}{2} \\ &\quad + \tau_w\lambda w\|u-c_1\|_2^2 + \tau_w\lambda(1-w)\|u-c_2\|_2^2 \\ &\quad + \chi_{[0,1]}(w) \\ &= \text{argmin}_w \frac{\|w-\tilde{w}\|_2^2}{2} \\ &\quad + \tau_w\lambda w(\|u-c_1\|_2^2 - \|u-c_2\|_2^2) \\ &\quad + \chi_{[0,1]}(w) \\ &= P_{[0,1]}\left(\tilde{w} - \tau_w\lambda(\|u-c_1\|_2^2 - \|u-c_2\|_2^2)\right).\end{aligned} \tag{28}$$ ∎

LEMMA 2.4. *The proximal operator of $h(u,w) + F(u)$ is:*

$$\text{prox}_{\tau_u(h(u,w)+F(u))}(\tilde{u}) = P_{\mathcal{R}}\left(\frac{\tilde{u} - \tau_u\lambda 2(wc_1 + (1-w)c_2)}{1+2\tau_u\lambda}\right), \tag{29}$$

where $P_{\mathcal{R}}$ is the pixel-wise projection onto the standard range for chrominances.

Proof. The function is

$\tau_u\lambda w\|u-c_1\|_2^2+\tau_u\lambda(1-w)\|u-c_2\|_2^2+\chi_{\mathcal{R}}(u)$:

$$\begin{aligned}
\operatorname{prox}_{\tau_u(h(u,w)+F(u))}(\tilde{u}) &= \operatorname{argmin}_u \|u-\tilde{u}\|_2^2+2\tau_u\lambda w\|u-c_1\|_2^2\\
&\quad+2\tau_u\lambda(1-w)\|u-c_2\|_2^2+\chi_{\mathcal{R}}(u)\\
&= \operatorname{argmin}_u \|u\|_2^2-2\langle u|\tilde{u}\rangle-4\tau_u\lambda\langle u|wc_1\rangle+2\tau_u\lambda w\|u\|_2^2\\
&\quad-4\tau_u\lambda\langle u|(1-w)c_2\rangle+2\tau_u\lambda(1-w)\|u\|_2^2+\chi_{\mathcal{R}}(u)\\
&= \operatorname{argmin}_u (2+2\tau_u\lambda)\|u\|_2^2-4\tau_u\lambda\langle u|wc_1+(1-w)c_2\rangle\\
&\quad-2\langle u|\tilde{u}\rangle+\chi_{\mathcal{R}}(u)\\
&= \operatorname{argmin}_u \|u\|_2^2-2\left\langle u\middle|\frac{\tilde{u}+\tau_u\lambda 2(wc_1+(1-w)c_2)}{2+2\tau_u\lambda}\right\rangle\\
&\quad+\chi_{\mathcal{R}}(u)\\
&= P_{\mathcal{R}}\left(\frac{\tilde{u}+\tau_u\lambda 2(wc_1+(1-w)c_2)}{1+\tau_u\lambda}\right). \qquad (30)
\end{aligned}$$

∎

The final algorithm, solving (1) for one frame reads as Algorithm 2. $P_{B(0,\alpha)}$ and $P_{B_{\mathbb{R}^P}(0,\beta)}$ represent the projection onto the L^2 ball of radius α and β respectively. $P_{[0,1]}$ is the truncation of the coordinates between 0 and 1. $P_{\mathcal{R}}$ is the projection onto the standard range for chrominances.

After iterating, the values of w are projected onto the canonical basis to have binary values and to avoid melting of colors or correspondence maps. A binarization of the variable w by truncation enables the computation of the correspondence map v. The algorithm computes two colors for each pixel: one corresponding to the final result u and another one, $\sum_i w_i c_i$ corresponding to the transfer of colors from the initial frame through the sequence.

A theoretical analysis of Algorithm 2 and its relationship with the PALM algorithm of [5] is studied in Appendix B. The convergence is numerically verified by computing the value of the functional during the iterations. The values are shown in Figure 5. We can see that the functional is decreasing and becomes asymptotically constant, which shows the numerical convergence.

3. Interactive Scribble-based Correction. As it is difficult to provide a fully automatic method for image colorization (*e.g.*, [28]), it is also hard to design an unsupervised frame-to-frame approach. Indeed, occlusions or dis-occlusions may occur, and new objects can appear in the video sequence whose color information is not present in the beginning of the sequence. For instance, Figure 15 shows a video sequence where the frame-to-frame propagation model is suitable. In contrast, in Figure 14, the method requires an intervention of the user to correct a dis-occlusion.

To cope with this limitation, we propose in this section a manual correction of the frames that extends the frame-to-frame approach. We also describe a variational framework for the scribble correction, as well as a minimization algorithm.

3.1. Overview of the User-guided Correction Method . To correct the visually unsuitable results of the proposed frame-to-frame propagation approach (1), we design a user-guided correction method widely inspired by the model proposed by Pierre *et al.* [28]. We extend this work to the case of videos, where 3D blocks are considered (2D + t). Frequently, occlusion or dis-occlusion parts of the video sequence produces undesirable results, which is not always easily visible on the first frame it occurs. Thus, in our method, the user can define the sub-sequence to correct.

Algorithm 2 Minimization of (23) with $\mathcal{M} = 2$.

Input: $v_{1,2}$, $c_{1,2}$

$w = 0.5$ and $u^0 = \sum_{i=1}^{2} w_i c_i$.

$p^0 \leftarrow \nabla u^0$

$z^0 \leftarrow A w^0$

for $n \geq 0$ **do**

$p^{n+1} \leftarrow P_{B(0,\alpha)}\left(p^n + \sigma_u \nabla \overline{u}^n\right)$

$z^{n+1} \leftarrow P_{B_{\mathbb{R}^P}(0,\beta)}(z^n + \sigma_w((v_1 - v_2) \otimes \nabla \overline{w}^n + (\nabla v_1 - \nabla v_2)\,\overline{w}^n + \sigma_w \nabla v_2))$

$$w^{n+1} \leftarrow P_{[0,1]}(w^n + \tau_w((\nabla v_1 - \nabla v_2) z^{n+1} + \operatorname{div}(I_2 \otimes (v_1 - v_2)^T v^{n+1}) - \tau_w \lambda(\|u - c_i\|)_i))$$

$$u^{n+1} \leftarrow P_{\mathcal{R}}\left(\frac{u^n + \tau_u\left(\operatorname{div}(p^{n+1}) + \lambda \sum_{i=1}^{2} w_i^{n+1} c_i\right)}{1 + \tau_u \lambda}\right)$$

$\overline{w}^{n+1} \leftarrow 2w^{n+1} - w^n$

$\overline{u}^{n+1} \leftarrow 2u^{n+1} - u^n$

end for

$$w_{\text{optimal}} \leftarrow \begin{cases} 0 \text{ if } w^\infty \leq 0.5 \\ 1 \text{ otherwise.} \end{cases}$$

Output: $\hat{u} \leftarrow u^\infty$;

$\hat{w} \leftarrow w^\infty$;

$\hat{v} \leftarrow \hat{w} v_1 + (1 - \hat{w}) v_2$.

The correction of a dis-occlusion artefact requires the computation of a color on an object that appears on the scene. Thus, this pixel cannot have a reliable correspondence map from the previous frame, so the computation of a correspondence map is unreliable in dis-occlusion areas.

The correction algorithm, through the minimization of a new functional, computes a trade-off between the previous propagation result and the color provided by the scribbles. Compared to existing manual methods [38], as we use the current colorization result, few scribbles are necessary. The user intervention is thus reduced. The overview of the correction method is presented in Figure 6.

3.2. User-guided Correction Model. Assume that a first result $\tilde{c}_1$ has been computed from Model (1), from time 1 to time n, and that the user adds S scribbles on the unsuitable result on the n-th frame, providing S candidates $\tilde{c}_i$, $i = 2 \dots S+1$. One candidate is providing for all the pixels of the video sequence.

The next step corrects the video sequence between time t_1 (defined by the user) and time n. Generally, the number of unsuitable frames is enough small (less than 5) to correct the result with a limited computation time.

In the same context, authors of [28] propose to unify a user-guided image colorization with an exemplar-based one within a variational framework. The following model, which is a direct extension of [28] to spatio-temporal blocks, is minimized,

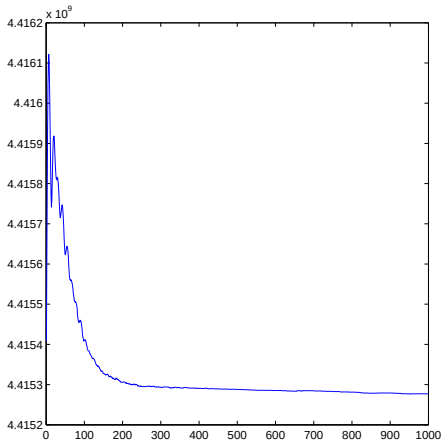


FIG. 5. *The values of the functional decrease during the iterations of Algorithm 2 and become asymptotically constant. This behavior highlights the numerical convergence of this algorithm.*

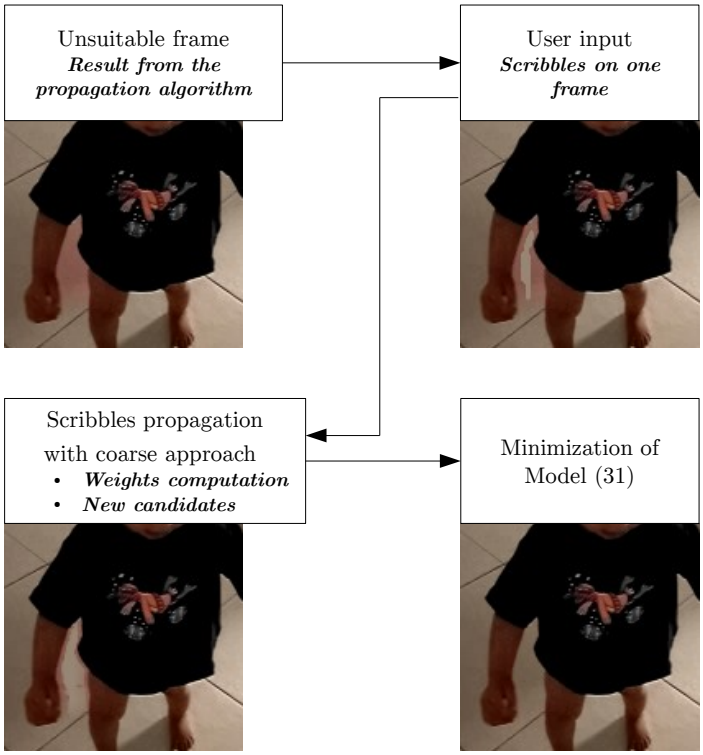


FIG. 6. *Interactive Scribble-based Correction. The correction algorithm provides a trade-off between the colors produced by the frame-to-frame propagation method, and the ones given by the user. A coarse approach (the geodesic distance diffusion for instance) is used to provide a first scribble diffusion, which is then refined by minimizing Functional* (31).

where u is a 2D+t chrominance block:

$$(\hat{u},\hat{w}) = \operatorname{argmin}_{u,w} \alpha\, \mathrm{TV}_{[t_1,n]}(u) + \lambda \int_{\Omega\times[t_1,n]} \sum_{i=1}^{S+1} w_i^{(t)}(x)\|u^{(t)}(x) - \tilde{c}_i^{(t)}(x)\|_2^2\, dx\, dt + \chi_{\mathcal{R}}(u) + \chi_{\mathcal{E}}(w). \tag{31}$$

The minimizer of the data-fidelity term

$$\int_{\Omega\times[t_1,n]} \sum_{i=1}^{S+1} w_i^{(t)}\|u^{(t)}(x) - \tilde{c}_i^{(t)}(x)\|_2^2\, dx\, dt, \tag{32}$$

with respect to u, is equal to the average of $\tilde{c}_i$ weighted by w_i. We add $\chi_{\mathcal{E}}(w)$, which constrains w onto the canonical basis. This constraint prevents the final result to be a melting of input colors.

The total variation (TV) of a chrominance block (U, V) reads:

$$\mathrm{TV}_{[t_1,n]}(u) = \int_{\Omega\times[t_1,n]} \left(\gamma\|\Lambda\nabla Y\|_2^2 + \|\Lambda\nabla U\|_2^2 + \|\Lambda\nabla V\|_2^2\right)^{\frac{1}{2}}, \tag{33}$$

with $\nabla = (\partial_x, \partial_y, \partial_t)$,

$$\Lambda := \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \mu \end{pmatrix}. \tag{34}$$

The temporal regularization is more important for Model (31) than for Model (1). Indeed, in Model (1), in the set of the initial correspondence maps, it may be possible to use temporally consistent maps, such as optical flow. In contrast, for the scribble correction model, the result is temporally regularized by adding a time derivative in the total variation term. The parameter μ controls the influence of the temporal regularization with respect to the spatial one.

3.3. Algorithm. For the minimization of Model (31), the algorithm presented in [27] is directly used. The composition of operator Λ with the gradient and the divergence is not restrictive to apply the general algorithm of [27]. Finally, the minimization of Model (31) is achieved with Algorithm 3, where c_*^n states for the closest candidate $\tilde{c}_i$ from u^n, and τ, σ are time steps. This Algorithm would contain an update for the variable w with the following form:

$$w^{n+1} \leftarrow \operatorname{argmin}_w \|w - w^n\|_2^2 + \rho \sum_{i=1}^{S+1} w_i\|u^n - \tilde{c}_i\|_2^2 + \chi_{\Delta}(w). \tag{35}$$

The convergence of such implicit numerical scheme does not depend on the ρ variable and it is pushed forward to $+\infty$. The line (35) becomes:

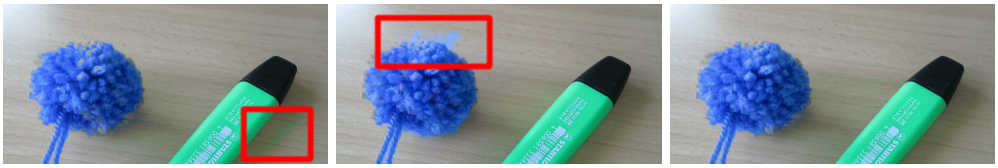
$$w^{n+1} \leftarrow \operatorname{argmin}_w \sum_{i=1}^{S+1} w_i\|u^n - \tilde{c}_i\|_2^2 + \chi_{\Delta}(w), \tag{36}$$

and it is minimised by $w^* = (0, \ldots, 1, 0, \ldots, 0)$ with 1 in the i-th position, where $\|u^n - c_i\|$ is the lowest one. Thus $\sum_{i=1}^{S+1} w^* c_i = c_j$ with c_i the closest candidate from u^n and denoted by c_*^n.

Algorithm 3 Minimization of (31).

$u^0 = \sum_{i=1}^{S+1} w_i c_i$.
$p^0 \leftarrow \nabla u$
for $n \geq 0$ **do**
  $p^{n+1} \leftarrow P_{B(0,\alpha)}\left(p^n + \sigma\Lambda\nabla\overline{u}^n\right)$
  $u^{n+1} \leftarrow P_{\mathcal{R}}\left(\dfrac{u^n + \tau\left(\mathrm{div}(\Lambda p^{n+1}) + 2\lambda c_*^n\right)}{1 + 2\tau_u\lambda}\right)$
  $\overline{u}^{n+1} \leftarrow 2u^{n+1} - u^n$
end for

With the combination of Algorithm 2 for color propagation and Algorithm 3 for scribble correction, we propose, in this work, a complete framework for video colorization. It is composed of a frame-to-frame propagation approach, which is unsupervised, and a correction step when occlusions or dis-occlusions occur. The interaction between the two models is detailed in the next section.



(a) PatchMatch propagation. (b) TVL1 optical flow propagation. (c) Propagation result with Model (1).

FIG. 7. *Combination of PatchMatch and optical flow correspondence maps. The PatchMatch (resp. TVL1) propagation of colors is illustrated in (a) (resp. (b)). The result with Model* (1) *is illustrated in (c). By combining the advantages of both maps, the frame-to-frame propagation Model* (1) *performs a visually suitable result, whereas the initial maps produces visually unsuitable parts (see in red boxes).*

4. Implementation Details. In this section, we describe implementation details. In particular, we present the computation of the correspondence maps and the initialisation of the correction algorithm. We also explain how the two models for propagation and correction interact with each other.

4.1. Global Workflow of the Proposed Video Colorization Method. Currently, many state-of-the-art methods are applied to a 3D representation of the video (2D + t). The frame-to-frame propagation Model (1) could be extended to 3D representation where the whole video sequence is processed in a single shot. In the case of manual intervention, the whole video sequence needs to be computed again. By working frame-by-frame, the speed-up enables us to provide an interactive approach. Indeed, if the results are visually unsuitable and unexpected, this is visible after a short processing time. The user can add scribbles after a reduced waiting time.

In our approach, we assume that one frame of the grayscale sequence is colorized (for instance by [29]) and the colors are propagated to the adjacent ones, until the whole video sequence is colorized. The required intervention of the user is simple: he can check the solution; if the frame is unsuitable, he can add scribbles on the result and the correction Algorithm 3 is applied. In practical cases the frame-to-frame

propagation efficiency reduces user's interventions. This efficiency is demonstrated in Section 5.

The global workflow is illustrated in Figure 1. The video is colorized by the propagation algorithm, until an unsuitable result can appear (*i.e.*, because of an occlusion). The user adds scribbles to correct the last unsuitable frame and the proposed correction method described in Section 3 computes a solution for the subset of frames. After that, the frame-to-frame propagation method can be re-applied from the last colorized frame.

4.2. Computation of the Correspondence Maps. Our method propagates colors from an initial frame to the final one, through an optimal correspondence map that is computed from initial correspondence maps with Equation (1). In this work, we compute two initial maps, from the luminance channel of the video sequence, with the PatchMatch method [3] and the TVL1 optical flow [7].

The PatchMatch method consists in computing a nearest-neighbor map between two adjacent frames. For each pixel at position x in the frame $u^{(t)}$ at time t, the nearest-neighbor map consists in the position $\hat{y}$ in the frame $u^{(t-1)}$ at time $t-1$ such that:

$$\hat{y} = \operatorname{argmin}_{y\in\Omega} \|P(x) - P(y)\|_2, \tag{37}$$

where $P(x)$ (resp. $P(y)$) is the patch centred at position x (resp. y) in frame $u^{(t)}$ (resp. $u^{(t-1)}$). The computation of this correspondence map through exhaustive search being extremely computationally expensive, PatchMatch algorithm [3] is used to compute an approximate correspondence map between two adjacent frames.

A second correspondence map is computed by the TVL1 optical flow with the algorithm of Chambolle-Pock [7]. The optical flow estimates the apparent motion between two frames of a video sequence. This estimation solves the brightness constancy constraint [36]:

$$\rho(u, v) = \partial_t u + \langle \nabla_{x,y} u | v \rangle + \beta u, \tag{38}$$

where $\nabla_{x,y} u$ is the spatial gradient and $v : \Omega \to \mathbb{R}^2$ is the motion field. TV is defined as in Equation (7). The following functional is minimized:

$$\min_v \mathrm{TV}(v) + \mathrm{TV}(u) + \|\rho(u, v)\|_1, \tag{39}$$

which gives the TVL1 optical flow.

These two correspondence maps, one computed by PatchMatch and the other one from TVL1 optical flow differ in term of quality: the first one is more piece-wise, and the second one is more regular. The PatchMatch algorithm, being less regularized, produces some colorization artefacts that are propagated over time. Nevertheless, it can deal with large displacements. In contrast, the regularity of the TVL1 map enforces the suitability of the estimated flow, but the brightness constancy constraint may not be reliable on some part like dis-occlusions.

The approximate nearest-neighbor map provided by the PatchMatch method is experimentally piece-wise constant. This geometrical property enables to simulate a copy-paste technique. Thus, the combination of the optical flow with the PatchMatch correspondence map can be understood as follows: for small and controlled motions, the TVL1 optical flow provides a suitable colorization; in the case of large displacements or dis-occlusions, PatchMatch copy-pastes colorized parts from another area of the image.

Figure 7 shows the differences of visually unsuitable results produced by PatchMatch and the TVL1 optical flow (red boxes focus on the problems). Model (1) produces a visually suitable result from these two initial maps.

4.3. Scribble Correction with Geodesic Distance. With the correction Model (31), we compute a solution from the frame-to-frame propagation result and user's scribbles. This model performs a scribble diffusion thanks to the total variation minimization. To help the propagation of the scribbles, a technique inspired by [38] is used. The geodesic distance is computed on a three dimensional block, composed of the two spatial dimensions with the temporal one. This geodesic distance provides, for each pixel, a weight w_i associated to the chrominance candidate $\tilde{c}_i$ given by one of the scribbles. $\tilde{c}_1$ is the previous frame-to-frame propagation result, $\tilde{c}_i$ with $i = 2, \dots, S+1$ is the chrominance given by one scribble.

In practical cases, the three dimensional gradient norm is filtered by a Gaussian kernel. The geodesic distance map, denoted by D, is computed with the fast marching algorithm [32], with potential:

$$\left(\theta + \|\nabla u\|_2^2\right)^{-r}, \tag{40}$$

inspired by [8]. θ and r are shape parameters. D is normalized with an affine mapping to have a range between 0 and 1. We use the implementation of [26] to compute the geodesic distance.

The weights w_i of Algorithm 3, corresponding to the scribble candidates $\tilde{c}_i$, are initialized with $1 - D_i$ where D_i is the geodesic distance from the scribble. The variable w_1, corresponding to the candidate $\tilde{c}_1$ coming from the previous frame-to-frame propagation model, is initialized with $1 - \sum_{i=2}^{S+1} w_i$. When $\sum_{i=2}^{S+1} w_i > 1$, the weights are divided by this sum to avoid that $w_1 < 0$. If the geodesic distance is low, pixels get the color of the scribble. In contrast, if the geodesic distance is high, they are not influenced by the scribble and they get the color of the previous frame-to-frame propagation result.

Algorithm 3 is pixel-wise initialized with $\sum_{i=1}^{S+1} w_i \tilde{c}_i$ to favor the scribbled result. Model (31) being non-convex, the result of the minimization directly depends on the initialization of Algorithm 3. Thanks to the diffusion by the geodesic distance, the iterative Algorithm 3 begins closer to the desired result and is thus faster. The geodesic distance computes a coarse but fast propagation. In contrast, the minimization of Functional (31) is slower, but produces a finer result.

Figure 6 illustrates the correction workflow. The geodesic distance propagates the color of the scribble. The minimization of Model (31) refines this first propagation that can be re-used by the frame-to-frame propagation algorithm since the result is visually suitable for the user.

4.4. Parameter Settings. The results presented in Section 5 have been performed with the following set of parameters: $\lambda = 0.1$, $\alpha = 25$ and $\gamma = 35$, in Equations (1) and (31), $\beta = 0.1$ in Equation (1), $\mu = 0.1$ in Equation (31), and $\theta = 45$, $r = 8$ in Equation (40). These parameters are the same for all the videos. We consider 2000 iterations for Algorithms 2 and 3, that are computed in about 1 second per frame at image resolution 352×256 with an unoptimized NVIDIA CUDA implementation. This computation time is about the same for the propagation step and the correction one. It is fast enough to consider the user-guided interaction.

5. Numerical Results. In this section, we compare our approach with state-of-the-art methods. First, we compare our approach with [38] and [22]. Next, we

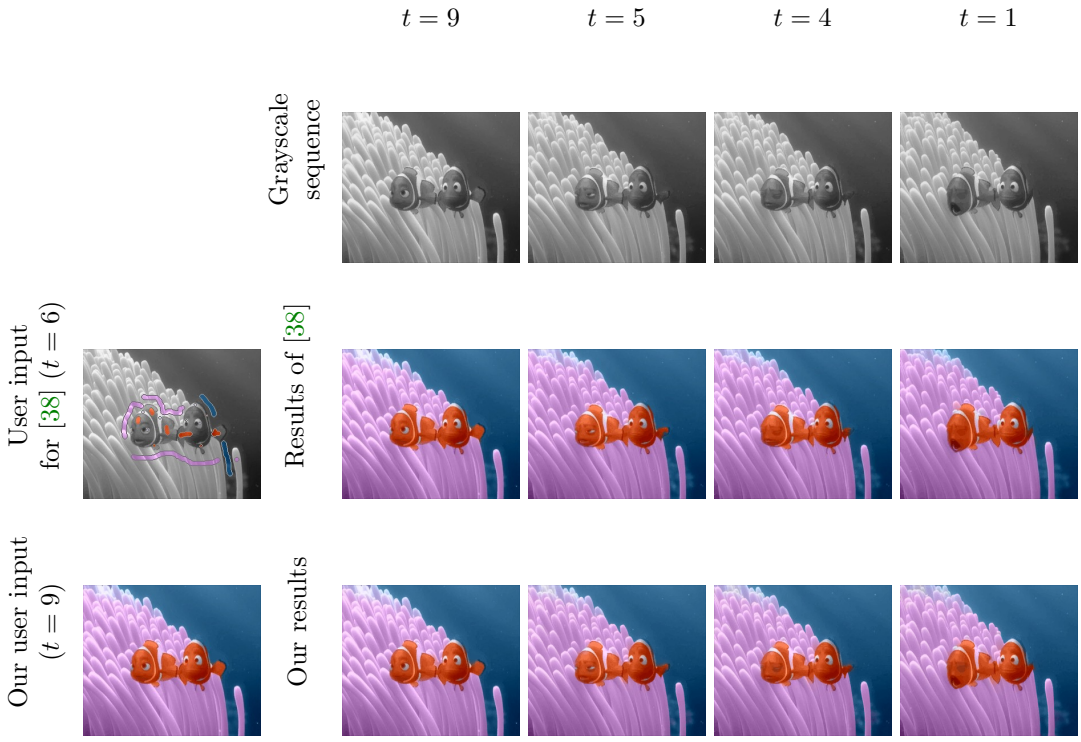


FIG. 8. *Comparison with [38] on the Nemo video sequence. We use the last colorized frame given by the method of [38] and we propagate its colors. No scribble correction has been used for our results. With our method, the fishtail is better colorized, zooms are available in Figure 9.*

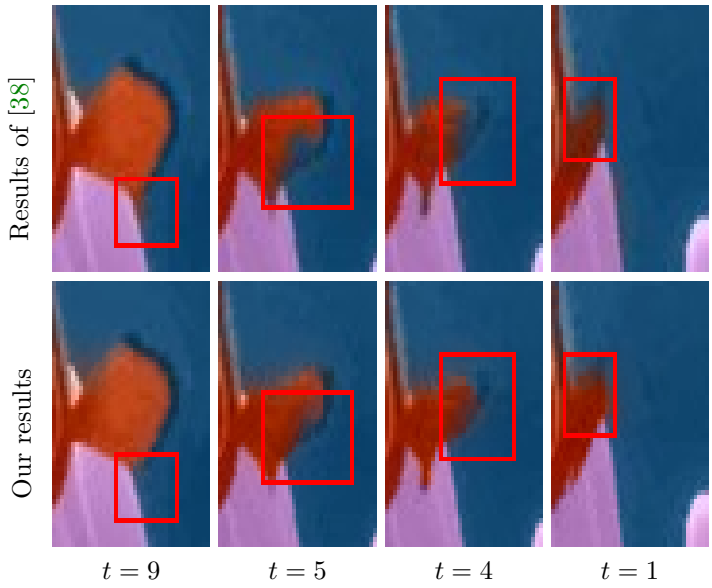


FIG. 9. *Zooms on Figure 8. The contours are better preserved with our approach (see, e.g., contours in the red boxes).*

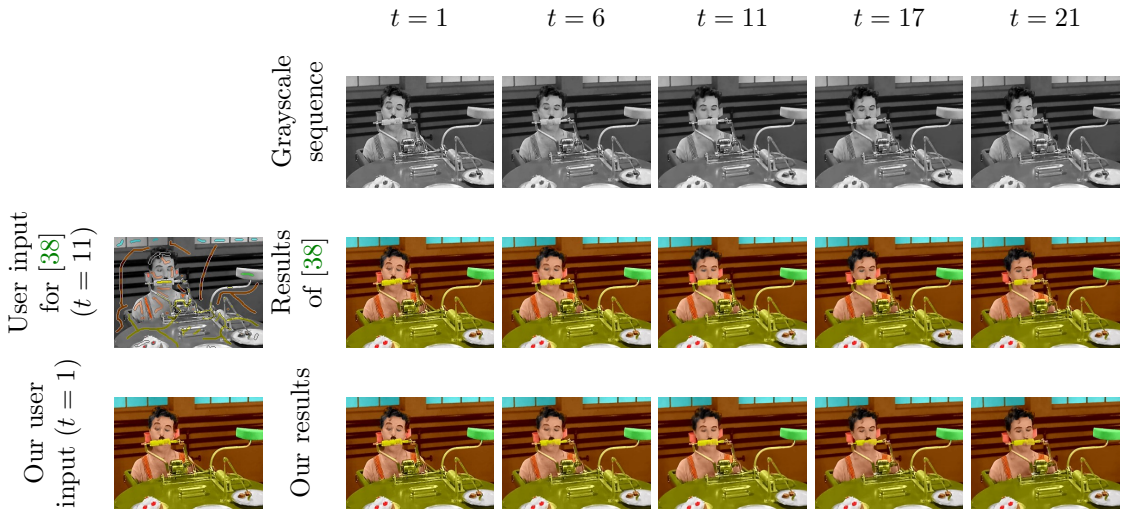
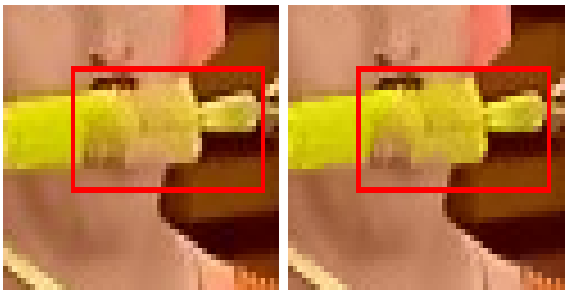


FIG. 10. *Comparison with [38] on the Chaplin video sequence. We use the first colorized frame given by the method of [38] and we propagate its colors. No scribble correction has been used for our results. With our method, the corncob is better colorized, zooms are available in Figure 11.*



Result of [38] at $t = 21$. Model (1) at $t = 21$.

FIG. 11. *Zooms on Figure 10. In the red box, the method of [38] mixes the yellow of the corncob with the color of the skin. With our method, the colors of the initial frame are less blended and the contours better preserved.*

propose some experiments on historical grayscale video sequences whose colorization is more challenging.

The data used for experiments have been taken from videos available on authors' websites, which contain noise from compression artefacts. Even with this low quality, our method performs promising results, which shows its robustness to the noise.

5.1. Comparison with Chrominance Blending [38]. First, let us compare our method to [38]. The two videos tested in Figures 8 and 10 have been taken from [38]. The video sequences *Nemo* and *Chaplin* are colorized by a propagation of one frame. The initial colored frame has been taken from the results of [38].

Figure 8 (resp. Figure 10) shows frames of the results from [38] and the colorization performed with our propagation algorithm on the video *Nemo* (resp. on the video *Chaplin*).

For the video *Chaplin*, the results produced with our method are comparable with the one of [38]. There are no visible artifacts and the final palette of colors is visually

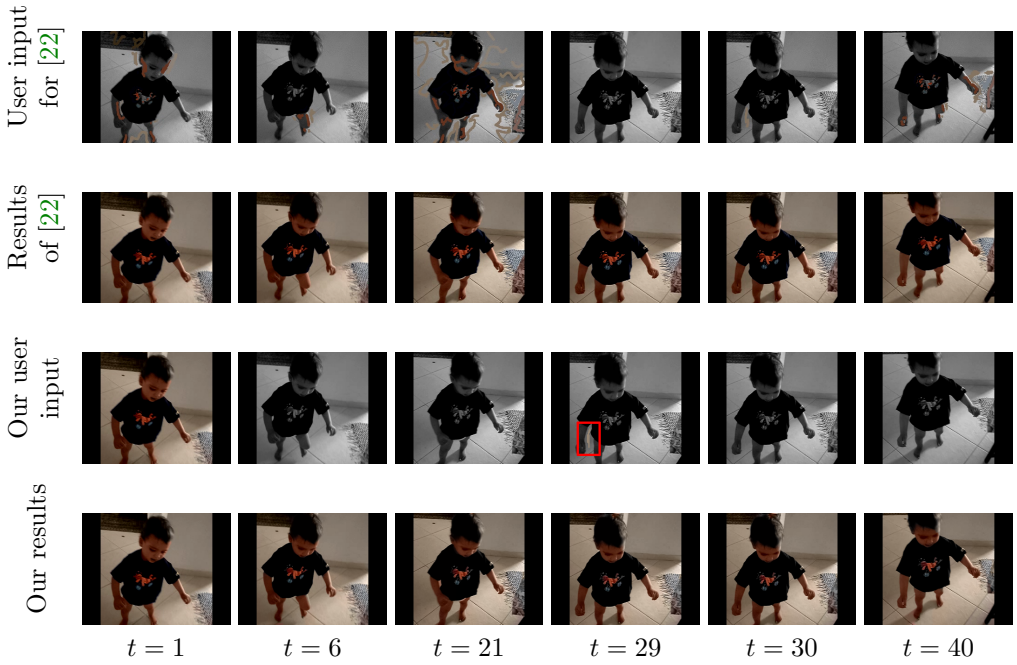


FIG. 12. *Comparison with [22] on the Toddler video sequence. We use the first colorized frame given by the method of [22] and we propagate its colors. Our method requires one scribble at time* $t = 29$ *(visible in the red box). Details of the correction step are available in Figure 6.*

the same.

For the *Nemo* video sequence, the fishtail is better colorized with our method. Figure 9 shows zooms of Figure 8. With our method, the contours are better preserved (see in red boxes) thanks to the coupled total variation (2). The method of [38] produces an artefact at the bottom of the tail on the last frame (an orange scratch blows in the background). Although our method propagates result of [38], it corrects this visually unsuitable result automatically, thanks to the total variation regularization.

Figure 10 presents experiments on the *Chaplin* video sequence. The frame-to-frame propagation is used without scribble correction. In Figure 11 we can see zooms on Figure 10. In the red box, after 20 frame-to-frame propagations, the method of [38] mixes the yellow color with the background. In contrast, with our method, the contours are better preserved.

As a conclusion of these experiments, our method is suitable on video sequences with large and constant parts.

5.2. Comparison with Quadratic Optimization [22]. Now, let us compare our method on examples taken from [22].

These examples are difficult to process with a frame-to-frame model because some new objects appear in the scene, and occlusions and dis-occlusions occur. In these videos, it is required to add scribbles to some frames of the sequence to correct visually unsuitable results of our frame-to-frame propagation algorithm. The method of [22] being based on a spatio-temporal processing of the sequence, it is naturally able to deal with occlusion problems. We show that our method tackles this issue with a small number of scribbles.

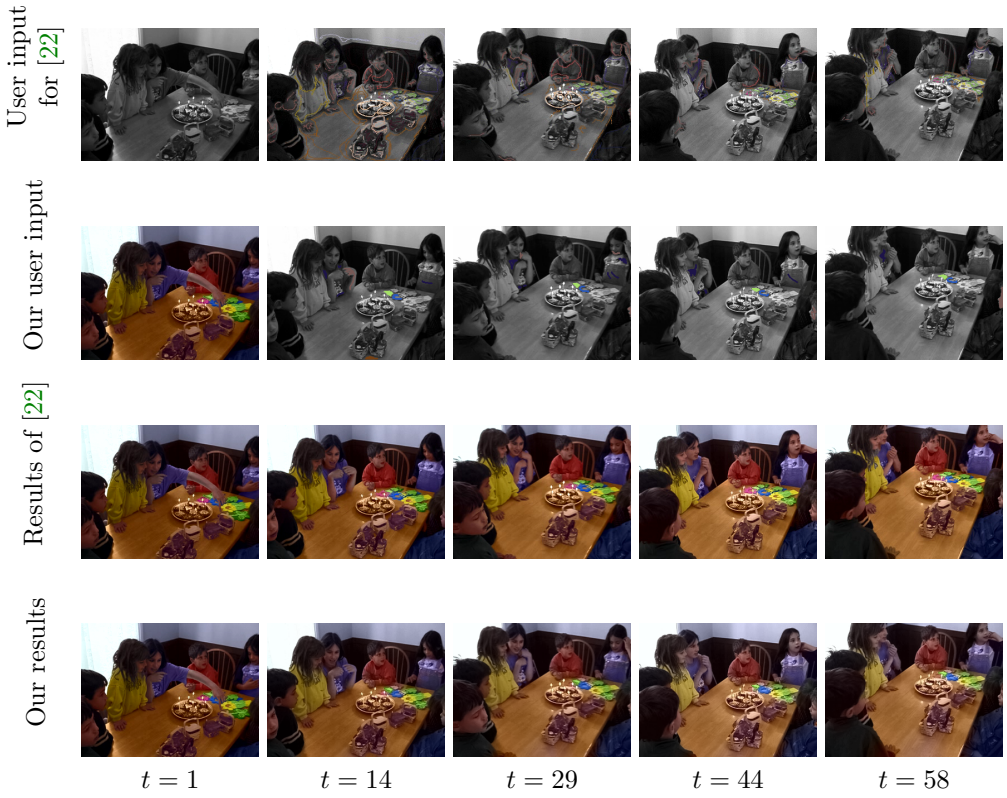


FIG. 13. *Comparison of our method with [22] for the Birthday video sequence. We use the first colorized frame given by the method of [22] and we propagate its colors. For our method, the user draws about 50 scribbles to correct the propagation algorithm.*

Figure 12 shows the result of our method in comparison with [22]. The results are visually the same. For the method of [22], it is required to add scribbles on some key frames of the video, and the result is computed in one step with no more interaction. To ensure to use the same color set, we use the first frame of the colorized result of [22] as initial frame for the frame-to-frame propagation Model (1). On this sequence we only use one scribble to correct the result (in the red box). In contrast, the method of [22] requires more than 50 scribbles.

Figure 13 shows another comparison with [22]. The initial color frame used by our algorithm comes from the result of [22]. In this example, we use about 50 scribbles to correct the unsuitable propagation results, whereas the method of [22] uses hundreds of it.

Moreover, with our approach, it is not needed to process and the user to check the whole video sequence after adding scribble, because the correction algorithm only focuses on few frames.

5.3. Experiments on Historical Videos. In this section, we consider the challenging problem of colorization of historical videos. The difficulties come from the noise, the flickering effect and the scratches.

First, we perform the colorization of a textured historical sequence. The first

FIG. 14. *Result of our method on the De Gaulle video sequence from the French* Institut National des Archives. *The initial frame ($t = 1$) is obtained by the unified method of Pierre et al. [28]. The colors of the initial frame are then propagated over the whole video sequence with only one scribble correction, drawn on the thumb into the red box, on the frame at $t = 25$.*

frame is colorized with [29]. In the *De Gaulle* video sequence, the thumb is hidden at the beginning of the sequence before appearing. Thus, it is unsuitably colorized by the propagation algorithm. We add a scribble onto the thumb when it appears (in the red box) and the correction algorithm integrates it. Finally, the propagation algorithm computes the colorization of the sequence. The 50 frames with image resolution of 352×256 pixels are colorized in about 1 min with our GPU implementation.

Finally, we apply our algorithm to the sequence *Pieds Nickelés* that is a very noisy video (Figure 15). This sequence is processed with the method of [12] to remove flicker effect. It contains a residual flicker effect as well as many scratches. Moreover, some of the displacements are very large, for instance, the heads of the characters. In spite of these challenging problems, our method is able to propagate colors over the 100 frames of the whole sequence without any correction. The grayscale sequence of our method is available on the first and the third rows, whereas the result is on the second and fourth ones. The initialization is the initial frame at $t = 1$.

The complete video colorization results are available online at:
http://www.labri.fr/perso/fpierre/video/video.html.

6. Conclusion. In this paper, we have proposed a novel variational model able to merge multiple correspondence maps to provide a suitable frame-to-frame propagation algorithm. A correction model integrating user scribbles is also proposed.

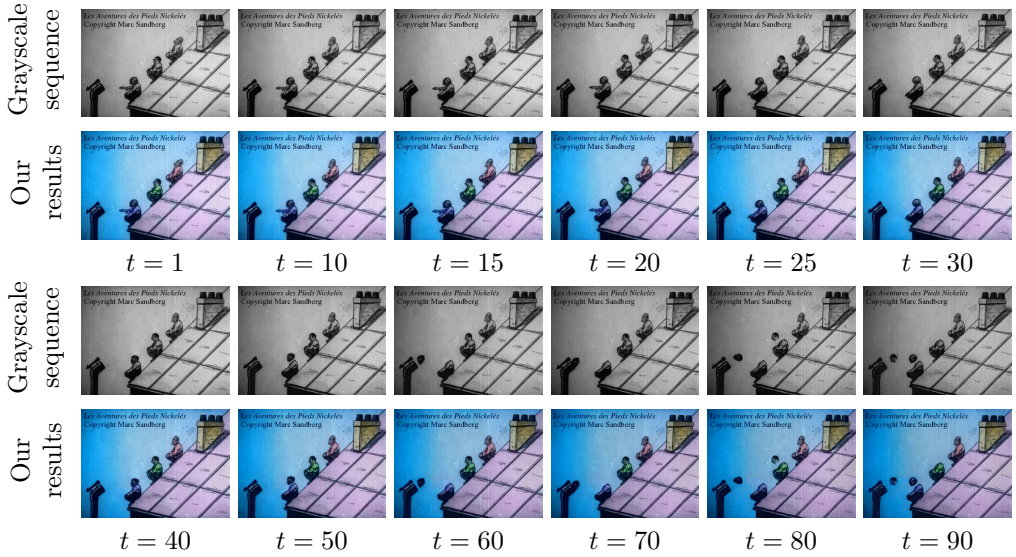


FIG. 15. *Result of our method for the Pieds Nickelés [12] video sequence. The initialization is the first frame at time $t = 1$, colorized with the method of [28]. For this sequence, besides the colorization of the first frame, no scribble has been added. This experiment shows that our approach is reliable on scratched videos with large displacements.*

These models are solved by primal-dual like algorithms integrated into an interactive framework. With our GPU implementation, we reach a high enough speed to consider an interactive software. Comparisons with state-of-the-art methods show the efficiency of our method and its ability to consider historical videos. As a future work, a data-fidelity term with L^1-norm could be considered to better preserve the color set.

Appendix A. Computation of the Map Operator.

In this section, we propose a proof of Lemma 2.1, which states an upper bound for the value of the operator norm of A.

Proof. (of Lemma 2.1) First of all, by triangular inequality:

$$\|A\| \leq \|A_1\| \, \|\nabla\| + \|A_2\| \,. \tag{41}$$

Let us compute an upper bound for each term in (41). It is known (see, *e.g.*, [6]) that $\|\nabla\|^2 = 8$.

Let us compute $\|A_1\|$:

$$\begin{aligned} \|A_1\| = \|A_1^*\| &= \left\| \begin{pmatrix} \varphi_1 - \varphi_2 & 0 & \psi_1 - \psi_2 & 0 \\ 0 & \varphi_1 - \varphi_2 & 0 & \psi_1 - \psi_2 \end{pmatrix} \right\| \\ &\leq \left\| \begin{pmatrix} \varphi_1 - \varphi_2 & 0 \\ 0 & \varphi_1 - \varphi_2 \end{pmatrix} \right\| + \left\| \begin{pmatrix} \psi_1 - \psi_2 & 0 \\ 0 & \psi_1 - \psi_2 \end{pmatrix} \right\| \\ &= |\varphi_1 - \varphi_2| + |\psi_1 - \psi_2|. \end{aligned} \tag{42}$$

The last quantity can be computed explicitly from the input data. This term can be bounded:

$$\|A_1\| \leq 2(N + M). \tag{43}$$

Let us compute $\|A_2\|$:

$$\begin{aligned}\|A_2\| = \|A_2^*\| = \big\|\big(&\partial_x\varphi_1 - \partial_x\varphi_2 \qquad \partial_y\varphi_1 - \partial_y\varphi_2 \\ &\partial_x\psi_1 - \partial_x\psi_2 \qquad \partial_y\psi_1 - \partial_y\psi_2\big)\big\| \\ &\leq |\partial_x\varphi_1 - \partial_x\varphi_2| + |\partial_y\varphi_1 - \partial_y\varphi_2| \\ &\qquad + |\partial_x\psi_1 - \partial_x\psi_2| + |\partial_y\psi_1 - \partial_y\psi_2|.\end{aligned} \tag{44}$$

This upper bound can be explicitly computed. This term can be bounded thanks to the bounding of the partial derivatives in Equation (44):

$$\|A_2\| \leq 4(N+M), \tag{45}$$

where N and M are the height and width of the frame.

Finally, a bounding of A reads as:

$$\|A\| \leq 2\sqrt{8}(N+M) + 4(N+M). \tag{46}$$

The operator norm depends linearly on the size of the frame. ∎

Appendix B. Link between Algorithm 2 and PALM Algorithm of [5].

The PALM algorithm proposed in [5] can be applied to Model (1). Indeed, this model can be written as

$$\min_{u\in\mathbb{R}^U, w\in\mathbb{R}^W} F(u) + h(u,w) + H(w) \tag{47}$$

with the same hypothesises as in Equation (16) for F and H, and h differentiable with respect to each variables with Lipschitz gradient. The PALM is recalled in Algorithm 4, where $\tilde{\sigma}_n$ and $\tilde{\tau}_n$ are time steps. To ensure the convergence to a critical point, the following bounds have to be fulfilled:

$$\begin{cases} \tilde{\sigma}_n < \dfrac{\gamma_1}{L_1(w^n)} \\ \tilde{\tau}_n < \dfrac{\gamma_2}{L_2(u^{n+1})}, \end{cases} \tag{48}$$

with $\gamma_1 < 1$, $\gamma_2 < 1$, $L_1(w^n)$ is the Lipschitz constant of $\nabla_u h(u, w^n)$ and $L_2(u^{n+1})$ is the Lipschitz constant of $\nabla_w h(u^{n+1}, w)$.

To apply the PALM to (1), we propose to make the following identifications:

- $F(u) = \alpha\,\mathrm{TV}_{\mathfrak{C}}(u) + \chi_{\mathcal{R}}(u)$,
- $h(u,w) = \dfrac{\lambda}{2}\int_\Omega \sum_{i=1}^{\mathcal{M}} w_i(x)\|u(x) - c_i(x)\|_2^2\,dx$,
- $H(w) = \beta\,\mathrm{TV}(v^{(t)}) + \chi_\Delta(w^{(t)})$.

Let us remark that, in this case, $L_1(w^n)$ and $L_2(u^{n+1})$ do not depend on n.

Algorithm 4 PALM Algorithm of [5].

for $n \geq 0$ **do**
  $u^{n+1} \leftarrow \mathrm{prox}_{\tilde{\sigma}_n F}\left(u^n - \tilde{\sigma}_n \nabla_u h(u^n, w^n)\right)$
  $w^{n+1} \leftarrow \mathrm{prox}_{\tilde{\tau}_n H}\left(w^n - \tilde{\tau}_n \nabla_w h(u^{n+1}, w^n)\right)$
end for

The proximal operators of H and G can be computed with the iterative primal-dual algorithm of Chambolle and Pock [7]. Indeed, let us compute $\mathrm{prox}_{\tilde{\sigma}F}(\tilde{u})$ in Algorithm 5. The time steps of this iterative algorithm are $\tilde{\sigma}_u$ and $\tilde{\tau}_u$.

Algorithm 5 Computation of $\text{prox}_{\tilde{\sigma}F}(\tilde{u})$ with the algorithm of [7].

for $n \geq 0$ **do**
  $p^{n+1} \leftarrow P_{B(0,\alpha)}\left(p^n + \tilde{\sigma}_u \nabla \overline{u}^n\right)$
  $u^{n+1} \leftarrow P_{\mathcal{R}}\left(\dfrac{u^n + \tilde{\tau}_u\left(\text{div}(p^{n+1}) + \tilde{\sigma}\tilde{u}\right)}{1 + \tilde{\tau}_u\tilde{\sigma}}\right)$
  $\overline{u}^{n+1} \leftarrow 2u^{n+1} - u^n$
end for

Algorithm 6 Computation of $\text{prox}_{\tilde{\tau}H}(\tilde{w})$ with the algorithm of [7].

for $n \geq 0$ **do**

$$z^{n+1} \leftarrow P_{B_{\mathbb{R}^P}(0,\beta)}(z^n + \tilde{\sigma}_w((v_1 - v_2) \otimes \nabla \overline{w}^n + (\nabla v_1 - \nabla v_2)\,\overline{w}^n + \tilde{\sigma}_w \nabla v_2))$$

$$w^{n+1} \leftarrow P_{[0,1]}\left(\frac{1}{1 + \tilde{\tau}_w\tilde{\tau}}[w^n + \tilde{\tau}_w((\nabla v_1 - \nabla v_2)z^{n+1} + \text{div}(I_2 \otimes (v_1 - v_2)^T v^{n+1}) + \tilde{\tau}_w\tilde{\tau}\tilde{w})]\right)$$

  $\overline{w}^{n+1} \leftarrow 2w^{n+1} - w^n$
end for

Is the same way, we propose to compute $\text{prox}_{\tilde{\tau}H}(\tilde{w})$ with an iterative algorithm. The time steps of this iterative algorithm are $\tilde{\sigma}_w$ and $\tilde{\tau}_w$.

With these two algorithms, it is possible to apply the PALM algorithm [5] to Model (1), using inner loops. To this aim, let us compute $\nabla_u h(u,w)$ and $\nabla_w h(u,w)$:

$$\nabla_u h(u,w) = \nabla_u\left(\frac{\lambda}{2}\int_\Omega \sum_{i=1}^{\mathcal{M}} w_i \|u - c_i\| dx\right) \tag{49}$$

$$= \lambda\left(u - \sum_{i=1}^{\mathcal{M}} w_i c_i\right). \tag{50}$$

$$\nabla_w h(u,w) = \nabla_w\left(\frac{\lambda}{2}\int_\Omega \sum_{i=1}^{\mathcal{M}} w_i \|u - c_i\| dx\right) \tag{51}$$

$$= \lambda(\|u - c_i\|)_i. \tag{52}$$

The complete PALM algorithm to solve Model (1) is written in Algorithm 7. The following proposition states the link between Algorithm 2 and the PALM one [5].

PROPOSITION B.1. *Let us consider Algorithm 7 with only one iteration for the inner loops (i.e., Algorithm 5 and 6) and a common value for $\tilde{\sigma}$ and $\tilde{\tau}$. Let us choose $\tau_u = \dfrac{\tilde{\tau}_u}{1 + \tilde{\tau}_u\tilde{\sigma} - \tilde{\tau}_u\lambda\tilde{\sigma}^2}$, $\sigma_u = \tilde{\sigma}_u$, $\tau_w = \dfrac{\tilde{\tau}_w}{1 + \tilde{\tau}_w\tilde{\tau}}$ and $\sigma_w = \tilde{\sigma}_w$. Then, Algorithm 7 is the same as Algorithm 2.*

Algorithm 7 PALM Algorithm of [5] applied to Model (1).

for $k \geq 0$ **do**
  $\tilde{w} \leftarrow w^n - \tilde{\tau}\lambda(\|u^n - c_i\|)_i$
  for $n \geq 0$ **do**
    $z^{n+1} \leftarrow P_{B_{\mathbb{R}^P}(0,\beta)}(z^n + \tilde{\sigma}_w((v_1 - v_2) \otimes \nabla\overline{w}^n + (\nabla v_1 - \nabla v_2)\,\overline{w}^n + \tilde{\sigma}_w \nabla v_2))$

$$w^{n+1} \leftarrow P_{[0,1]}\left(\frac{1}{1+\tilde{\tau}_w\tilde{\tau}}[w^n + \tilde{\tau}_w((\nabla v_1 - \nabla v_2)z^{n+1} + \operatorname{div}(I_2 \otimes (v_1 - v_2)^T\, v^{n+1}) + \tilde{\tau}_w\tilde{\tau}\tilde{w})]\right)$$

    $\overline{w}^{n+1} \leftarrow 2w^{n+1} - w^n$
  end for
  $\tilde{u} \leftarrow u^n - \tilde{\sigma}\lambda\left(u^n - \sum_{i=1}^{\mathcal{M}} w_i^{n+1} c_i\right)$
  for $n \geq 0$ **do**
    $p^{n+1} \leftarrow P_{B(0,\alpha)}\left(p^n + \tilde{\sigma}_u \nabla\overline{u}^n\right)$
    $u^{n+1} \leftarrow P_{\mathcal{R}}\left(\dfrac{u^n + \tilde{\tau}_u\left(\operatorname{div}(p^{n+1}) + \tilde{\sigma}\tilde{u}\right)}{1 + \tilde{\tau}_u\tilde{\sigma}}\right)$
    $\overline{u}^{n+1} \leftarrow 2u^{n+1} - u^n$
  end for
end for

The choice of the common value for $\tilde{\sigma}$ and $\tilde{\tau}$ is not a restrictive hypothesis to prove the convergence of the PALM algorithm. Indeed, a bound for each time step is required to ensure the convergence (see, *e.g.*, (48)). Taking the minimum between them, the two bound conditions are fulfilled.

Proof. Assume now to use only one iteration for the internal loops of Algorithm 7. Thus, the line 9 can be included in line 12. In this case, line 12 becomes:

$$u^{n+1} \leftarrow P_{\mathcal{R}}\left(\frac{u^n + \tilde{\tau}_u\left(\operatorname{div}(p^{n+1}) + \tilde{\sigma}\left(u^n - \tilde{\sigma}\lambda\left(u^n - \sum_{i=1}^{\mathcal{M}} w_i^n c_i\right)\right)\right)}{1 + \tilde{\tau}_u\tilde{\sigma}}\right). \tag{53}$$

with simplification:

$$u^{n+1} \leftarrow P_{\mathcal{R}}\left(\frac{u^n(1 + \tilde{\tau}_u\tilde{\sigma} - \tilde{\tau}_u\lambda\tilde{\sigma}^2) + \tilde{\tau}_u\left(\operatorname{div}(p^{n+1}) + \lambda\tilde{\sigma}^2\sum_{i=1}^{\mathcal{M}} w_i^n c_i\right)}{1 + \tilde{\tau}_u\tilde{\sigma}}\right). \tag{54}$$

Choosing $\tau_u = \dfrac{\tilde{\tau}_u}{1 + \tilde{\tau}_u\tilde{\sigma} - \tilde{\tau}_u\lambda\tilde{\sigma}^2}$ and $\tilde{\lambda} = \lambda\tilde{\sigma}^2$, we have

$$\frac{1 + \tilde{\tau}_u\tilde{\sigma}}{1 + \tilde{\tau}_u\tilde{\sigma} - \tilde{\tau}_u\lambda\tilde{\sigma}^2} = 1 + \tilde{\lambda}\tau_u. \tag{55}$$

Thus Equation (54) becomes:

$$u^{n+1} \leftarrow P_{\mathcal{R}}\left(\frac{u^n + \tau_u\left(\operatorname{div}(p^{n+1}) + \lambda\sum_{i=1}^{\mathcal{M}} w_i^n c_i\right)}{1+\lambda\tau_u}\right), \tag{56}$$

which is the same computation as line 8 of Algorithm 2.

The line 2 of Algorithm 7 can also be included in line 5 and this last line becomes:

$$\begin{aligned} w^{n+1} \leftarrow P_{[0,1]}\Big(\frac{1}{1+\tilde{\tau}_w\tilde{\tau}}[w^n + \tilde{\tau}_w((\nabla v_1 - \nabla v_2)z^{n+1} \\ + \operatorname{div}(I_2 \otimes (v_1-v_2)^T v^{n+1}) + \tilde{\tau}_w\tilde{\tau}(w^n - \tilde{\tau}\lambda(\|u^n - c_i\|)_i))]\Big). \end{aligned} \tag{57}$$

With simplification:

$$\begin{aligned} w^{n+1} \leftarrow P_{[0,1]}(w^n + \frac{\tilde{\tau}_w}{1+\tilde{\tau}_w\tilde{\tau}}((\nabla v_1 - \nabla v_2)z^{n+1} \\ + \operatorname{div}(I_2 \otimes (v_1-v_2)^T v^{n+1}) - \frac{\tilde{\tau}_w\tilde{\tau}^2}{1+\tilde{\tau}_w\tilde{\tau}}\lambda(\|u^n - c_i\|)_i)) \end{aligned} \tag{58}$$

Choosing $\tau_w = \dfrac{\tilde{\tau}_w}{1+\tilde{\tau}_w\tilde{\tau}}$, and $\tilde{\lambda} = \lambda\tilde{\tau}^2$, thus Equation (58) is the same computation as line 7 of Algorithm 2. The choice $\tilde{\lambda} = \lambda\tilde{\tau}^2$ is compatible with $\tilde{\lambda} = \lambda\tilde{\sigma}^2$ if we assume that $\tilde{\tau} = \tilde{\sigma}$. That is not a restrictive hypothesis, since it is possible to use the same time step in the PALM algorithm, by choosing the minimum of them as a common value to ensure the convergence. ∎

As a conclusion, Algorithm 2, inspired by the primal-dual one of Chambolle and Pock [7], can be seen as the PALM algorithm of [5] where the proximal operator for the total variation would be computed by the Chambolle and Pock Algorithm itself with only one iteration. The advantage of the PALM algorithm is the theoretical guarantee of convergence to some critical point. For our algorithm, the decreasing of the energy cannot be proved since it is a saddle-point problem. Thus, the bare bones of the convergence proof of PALM cannot be adapted to Algorithm 2. However, Proposition B.1 explains the good behavior of Algorithm 2 whose convergence has been numerically verified (see, *e.g.*, Figure 5).

Acknowledgment. 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.

REFERENCES

[1] P. Arias, G. Facciolo, V. Caselles, and G. Sapiro, *A variational framework for exemplar-based image inpainting*, International Journal of Computer Vision, 93 (2011), pp. 319–347.
[2] J.-F. Aujol, S. Ladjal, and S. Masnou, *Exemplar-based inpainting from a variational point of view*, SIAM Journal on Mathematical Analysis, 42 (2010), pp. 1246–1285.
[3] C. Barnes, E. Shechtman, A. Finkelstein, and D. Goldman, *Patchmatch: a randomized correspondence algorithm for structural image editing*, in Transactions on Graphics, ACM, 2009, pp. 24–32.
[4] M. J. Black and P. Anandan, *The robust estimation of multiple motions: Parametric and piecewise-smooth flow fields*, Computer vision and image understanding, 63 (1996), pp. 75–104.

[5] J. Bolte, S. Sabach, and M. Teboulle, *Proximal alternating linearized minimization for nonconvex and nonsmooth problems*, Mathematical Programming, 146 (2014), pp. 459–494.
[6] A. Chambolle, *An algorithm for total variation minimization and applications*, Journal of Mathematical Imaging and Vision, 20 (2004), pp. 89–97.
[7] A. Chambolle and T. Pock, *A first-order primal-dual algorithm for convex problems with applications to imaging*, Journal of Mathematical Imaging and Vision, 40 (2011), pp. 120–145.
[8] T. F. Chan and L. A. Vese, *Active contours without edges*, IEEE Transactions on Image Processing, 10 (2001), pp. 266–277.
[9] G. Charpiat, M. Hofmann, and B. Schölkopf, *Automatic image colorization via multimodal predictions*, in European Conference on Computer Vision, Springer, 2008, pp. 126–139.
[10] P. L. Combettes and V. R. Wajs, *Signal recovery by proximal forward-backward splitting*, Multiscale Modeling & Simulation, 4 (2005), pp. 1168–1200.
[11] S. Conti, J. Ginster, and M. Rumpf, *A BV functional and its relaxation for joint motion estimation and image sequence recovery*, ESAIM: Mathematical Modelling and Numerical Analysis, 49 (2015), pp. 1463–1487.
[12] J. Delon and A. Desolneux, *Stabilization of flicker-like effects in image sequences through local contrast correction*, SIAM Journal on Imaging Sciences, 3 (2010), pp. 703–734.
[13] D. Fortun, P. Bouthemy, and C. Kervrann, *Aggregation of local parametric candidates with exemplar-based occlusion handling for optical flow*, Computer Vision and Image Understanding, (2015).
[14] M. Ghanbari, *The cross-search algorithm for motion estimation*, IEEE Transactions on Communications, 38 (1990), pp. 950–953.
[15] R. K. Gupta, A. Y.-S. Chia, D. Rajan, E. S. Ng, and H. Zhiyong, *Image colorization using similar images*, in ACM International Conference on Multimedia, 2012, pp. 369–378.
[16] J.-H. Heu, D.-Y. Hyun, C.-S. Kim, and S.-U. Lee, *Image and video colorization based on prioritized source propagation*, in IEEE International Conference on Image Processing, 2009, pp. 465–468.
[17] D.-Y. Hyun, J.-H. Heu, C.-S. Kim, and S.-U. Lee, *Prioritized image and video colorization based on gaussian pyramid of gradient images*, Journal of Electronic Imaging, 21 (2012), pp. 023027–1.
[18] R. Irony, D. Cohen-Or, and D. Lischinski, *Colorization by example*, in Eurographics Symp. on Rendering, vol. 2, Citeseer, 2005.
[19] V. G. Jacob and S. Gupta, *Colorization of grayscale images and videos using a semiautomatic approach*, in IEEE International Conference on Image Processing, 2009, pp. 1653–1656.
[20] M. Lang, O. Wang, T. Aydin, A. Smolic, and M. Gross, *Practical temporal consistency for image-based graphics applications*, in Transactions on Graphics, ACM, July 2012, pp. 34–42.
[21] C. Lannaud, *Fallait-il coloriser la guerre ?*, L'express, (2009). http://www.lexpress.fr/culture/tele/fallait-il-coloriser-la-guerre_789380.html.
[22] A. Levin, D. Lischinski, and Y. Weiss, *Colorization using optimization*, in Transactions on Graphics, ACM, 2004, pp. 689–694.
[23] B. D. Lucas, T. Kanade, et al., *An iterative image registration technique with an application to stereo vision.*, in International Joint Conference on Artificial Intelligence, vol. 81, 1981, pp. 674–679.
[24] B. S. Manjunath and W.-Y. Ma, *Texture features for browsing and retrieval of image data*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 18 (1996), pp. 837–842.
[25] Z. Pan, Z. Dong, and M. Zhang, *A new algorithm for adding color to video or animation clips*, UNION Agency-Science Press, 2004.
[26] G. Peyré, *Toolbox fast marching - a toolbox for fast marching and level sets computations*, 2008. http://www.mathworks.com.
[27] F. Pierre, J.-F. Aujol, A. Bugeau, N. Papadakis, and V.-T. Ta, *Luminance-chrominance model for image colorization*, SIAM Journal on Imaging Sciences, 8 (2015), pp. 536—-563.
[28] F. Pierre, J.-F. Aujol, A. Bugeau, and V.-T. Ta, *A unified model for image colorization*, in Color and Photometry in Computer Vision (ECCV Workshop), 2014, pp. 1–12.
[29] F. Pierre, J.-F. Aujol, A. Bugeau, and V.-T. Ta, *Colociel*. Dépôt Agence de Protection des Programmes N° IDDN.FR.001.080021.000.S.P.2016.000.2100, 2016. http://www.labri.fr/perso/fpierre/colociel.zip.
[30] M. H. Quang, S. H. Kang, and T. M. Le, *Image and video colorization using vector-valued reproducing kernel hilbert spaces*, Journal of Mathematical Imaging and Vision, 37 (2010), pp. 49–65.

826 [31] L. I. Rudin, S. Osher, and E. Fatemi, *Nonlinear total variation based noise removal algo-*
827 *rithms*, Physica D: Nonlinear Phenomena, 60 (1992), pp. 259–268.
828 [32] J. A. Sethian, *Level set methods and fast marching methods: evolving interfaces in compu-*
829 *tational geometry, fluid mechanics, computer vision, and materials science*, vol. 3, Cam-
830 bridge university press, 1999.
831 [33] B. Sheng, H. Sun, S. Chen, X. Liu, and E. Wu, *Colorization using the rotation-invariant*
832 *feature space*, IEEE computer graphics and applications, 1 (2011), pp. 24–35.
833 [34] D. Sỳkora, J. Buriánek, and J. Žára, *Unsupervised colorization of black-and-white cartoons*,
834 in Proceedings of the 3rd international symposium on Non-photorealistic animation and
835 rendering, ACM, 2004, pp. 121–127.
836 [35] S. Teng, Y. Shen, Z. Zhao, L. Li, and M. Cao, *An interactive framework for video coloriza-*
837 *tion*, in IEEE International Conference on Image and Graphics, 2013, pp. 89–94.
838 [36] A. Wedel, T. Pock, C. Zach, H. Bischof, and D. Cremers, *An improved algorithm for*
839 *TV-L1 optical flow*, in Statistical and Geometrical Approaches to Visual Motion Analysis,
840 Springer, 2009, pp. 23–45.
841 [37] T. Welsh, M. Ashikhmin, and K. Mueller, *Transferring color to greyscale images*, in Trans-
842 actions on Graphics, ACM, 2002, pp. 277–280.
843 [38] L. Yatziv and G. Sapiro, *Fast image and video colorization using chrominance blending*,
844 vol. 15, 2006, pp. 1120–1129.
845 [39] Z. Zhen, G. Yan, and M. Lizhuang, *An automatic image and video colorization algorithm*
846 *based on pattern continuity*, in International Conference on Audio, Language and Image
847 Processing, 2012, 2012, pp. 531–536.