



HAL
open science

Interactive Video Colorization within a Variational Framework

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

► **To cite this version:**

Fabien Pierre, Jean-François Aujol, Aurélie Bugeau, Vinh-Thong Ta. Interactive Video Colorization within a Variational Framework. SIAM Journal on Imaging Sciences, 2017. hal-01379702v2

HAL Id: hal-01379702

<https://hal.science/hal-01379702v2>

Submitted on 19 Dec 2016 (v2), last revised 29 May 2017 (v3)

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

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

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.

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

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

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

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

66 All these techniques are based on the minimization of a quadratic criterion com-
 67 puted on the whole video sequence.

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

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

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

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

86 To tackle these issues, the frame-to-frame approaches are presented in the follow-
 87 ing.

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

91 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	✓	✓	✓

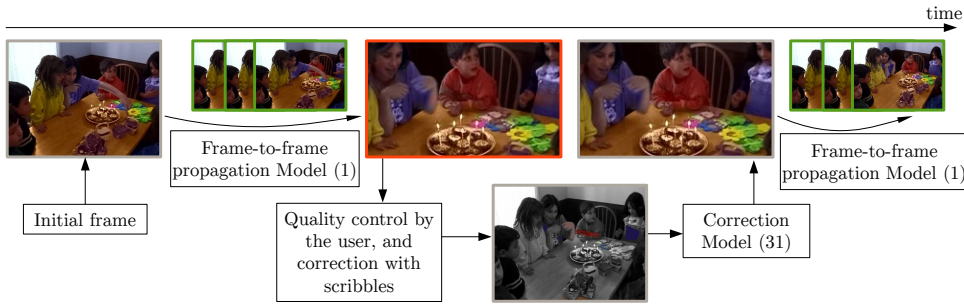


FIG. 1. Overview of the whole approach. For most of the sequence, the frame-to-frame propagation model computes suitable results (in green). In the case of dis-occlusion area, a user-guided approach is mixed with the unsuitable result (in red). After that, the frame-to-frame propagation restarts from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

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

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

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

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

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

144 **2.1. Overview of the Frame-to-Frame**

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

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

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

157 **2.2. A New Functional Based on Color Regularization and Correspon-**
 158 **dence Maps.** In this section, a model performing both a choice between \mathcal{M} corre-
 159 spondence maps and computing the color of a frame is proposed.

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

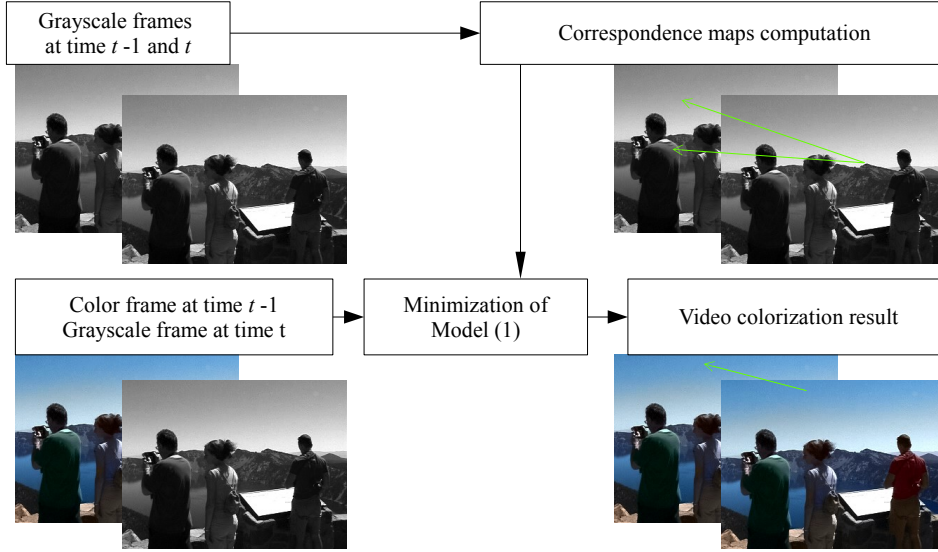


FIG. 2. Overview of the frame-to-frame propagation method. In this example, two correspondence maps between grayscale frames at times $t-1$ and t are computed in a first step. The frame-to-frame propagation method estimates both an optimal correspondence map and the final color for the frame at time 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.

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

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

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

172

$$173 \quad (1) \quad (\hat{u}^{(t)}, \hat{w}^{(t)}) = \operatorname{argmin}_{u^{(t)}, w^{(t)}} \alpha \operatorname{TV}_{\mathcal{C}}(u^{(t)}) + \beta \operatorname{TV}(v^{(t)})$$

$$174 \quad + \lambda \int_{\Omega} \sum_{i=1}^{\mathcal{M}} w_i^{(t)}(x) \|u^{(t)}(x) - c_i^{(t)}(x)\|_2^2 dx$$

175

$$+ \chi_{\mathcal{R}}(u^{(t)}) + \chi_{\Delta}(w^{(t)}).$$

177 $c_i^{(t)}$ is the color candidate given from the frame $u^{(t-1)}$, by the correspondence map
 178 $v_i^{(t)}$, thanks to its definition available in Equation (5) and (6). A trade-off between
 179 the regularity of the map $v^{(t)}$ and the colors of the result $u^{(t)}$ is controlled by the
 180 parameters α , β and λ . $w^{(t)}$ is a weight parameter that measures the pixel-wise
 181 contribution between the correspondence maps.

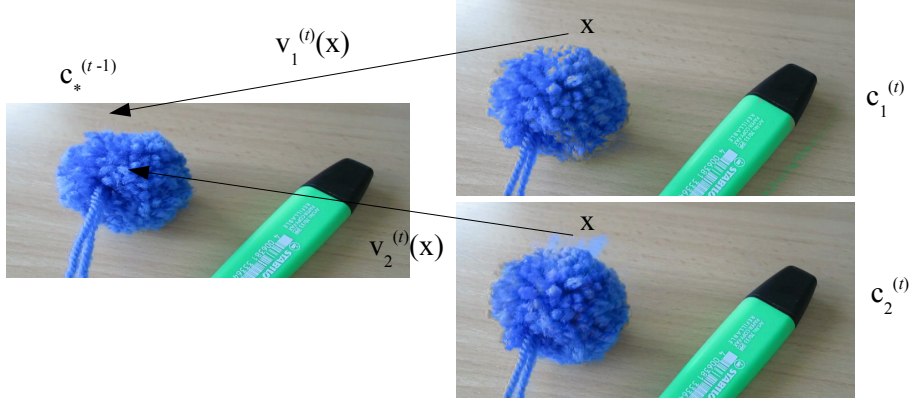


FIG. 3. Example of correspondence maps between two frames ($\mathcal{M} = 2$). From a given result at time $t - 1$, two possible propagation results are provided at time t from the two maps. The frame-to-frame model computes a regularized optimal final color.

182 $\text{TV}_{\mathfrak{C}}(u^{(t)})$ is a regularization term based on the minimization of the total varia-
 183 tion [31] of chrominance channels of the frame $u^{(t)}$ at time t :

$$184 \quad (2) \quad \text{TV}_{\mathfrak{C}}(u^{(t)}) = \int_{\Omega} (\gamma \|\nabla Y\|_2^2 + \|\nabla U\|_2^2 + \|\nabla V\|_2^2)^{1/2},$$

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

188 $\text{TV}(v^{(t)})$ enforces the regularity of the final correspondence map and is detailed
 189 in Section 2.3 (see Equation (7)). The correspondence map at time t denoted by $v^{(t)}$
 190 is defined as a weighted average of the $v_i^{(t)}$, $i = 1, \dots, \mathcal{M}$:

$$191 \quad (3) \quad v^{(t)} = \sum_{i=1}^{\mathcal{M}} w_i^{(t)} v_i^{(t)}.$$

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

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

$$197 \quad (4) \quad \hat{c}^{(t-1)} = \sum_{i=1}^{\mathcal{M}} \hat{w}_i^{(t-1)} c_i^{(t-1)}.$$

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

$$199 \quad (5) \quad c_i^{(t)}(x) = \hat{c}^{(t-1)}(v_i^{(t)}(x)), \quad i = 1, \dots, \mathcal{M},$$

200 and

$$201 \quad (6) \quad c_i^{(1)}(x) = u^{(1)}(v_i^{(1)}(x)), \quad i = 1, \dots, \mathcal{M}.$$

202 With these definitions, Functional (1) computes an optimal map between the frames
 203 $u^{(t-1)}$ and $u^{(t)}$ with a regularization of both maps and colors.

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

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

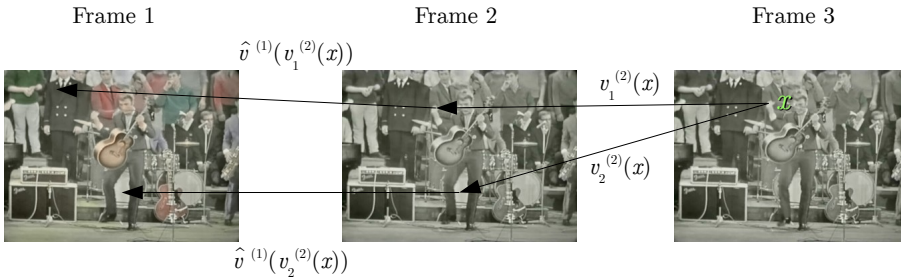


FIG. 4. Illustration of the color propagation from the first frame. The colors are transferred from frame 1 to frame 2 with respect to the optimal map $\hat{v}^{(1)} = \sum_{i=1}^{\mathcal{M}} \hat{w}_i^{(1)} v_i^{(1)}$ computed by the proposed frame-to-frame propagation algorithm. The data-fidelity 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.

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

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

224 **2.3. Correspondence Maps Regularization.** We propose to regularize both
 225 correspondence maps and colors of the result. Similar approach for map regulariza-
 226 tion is proposed by [13] to compute an aggregation of optical flows. In [13], the result
 227 is only based on the map regularization whereas our model considers both maps and
 228 chrominances regularity. In [11], a similar technique enables both the estimation of
 229 the optical flow and the denoising of the image. It differs from video colorization
 230 where the optical flow is computed on the luminance channel, whereas the final esti-
 231 mation is performed on chrominance channels. Other approaches preserving geometric
 232 structures have been proposed for inpainting problems [2, 1].

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

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

$$237 \quad (7) \quad \text{TV}(v) = \|\nabla v\|_{1,2} = \int_{\Omega} \|\nabla v(x)\|_2 dx.$$

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

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

$$250 \quad (8) \quad w_1 = w, w_2 = 1 - w,$$

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

252 With Parametrization (8):

$$253 \quad (9) \quad v = w(v_1 - v_2) + v_2.$$

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

$$255 \quad (10) \quad \nabla v = (v_1 - v_2) \otimes \nabla w + w(\nabla v_1 - \nabla v_2) + \nabla v_2,$$

256 where \otimes is the Kronecker product:

$$257 \quad (11) \quad (v_1 - v_2) \otimes \nabla w = \begin{pmatrix} (\varphi_1 - \varphi_2)\partial_x w \\ (\varphi_1 - \varphi_2)\partial_y w \\ (\psi_1 - \psi_2)\partial_x w \\ (\psi_1 - \psi_2)\partial_y w \end{pmatrix},$$

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

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

$$261 \quad (12) \quad Aw = \underbrace{\begin{pmatrix} \varphi_1 - \varphi_2 & 0 \\ 0 & \varphi_1 - \varphi_2 \\ \psi_1 - \psi_2 & 0 \\ 0 & \psi_1 - \psi_2 \end{pmatrix}}_{:=A_1} \nabla w + \underbrace{\begin{pmatrix} \partial_x \varphi_1 - \partial_x \varphi_2 \\ \partial_y \varphi_1 - \partial_y \varphi_2 \\ \partial_x \psi_1 - \partial_x \psi_2 \\ \partial_y \psi_1 - \partial_y \psi_2 \end{pmatrix}}_{:=A_2} w.$$

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

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

291 **2.4.2. Dual version of Model (1)** . To solve Model (1), the 1, 2–norm, recalled
 292 in Equation (7), is written in the dual form:

$$293 \quad (18) \quad \beta \|u\|_{1,2} = \max_{p \in \mathbb{R}^P} \langle u|p \rangle - \chi_{B_{\mathbb{R}^P}(0,\beta)}(p),$$

294 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.

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

$$\begin{aligned} 299 \quad \text{TV}(v) &= \|(v_1 - v_2) \otimes \nabla w + w(\nabla v_1 - \nabla v_2) + \nabla v_2\|_{1,2} \\ 300 &= \max_{z \in \mathbb{R}^{N \times 4}} \langle (v_1 - v_2) \otimes \nabla w + w(\nabla v_1 - \nabla v_2) + \nabla v_2 | z \rangle \\ 301 &\quad - \chi_{B_{\mathbb{R}^P}(0,\beta)}(z) \\ 302 \quad (19) &= \max_{z \in \mathbb{R}^{N \times 4}} \langle Aw | z \rangle + \langle \nabla v_2 | z \rangle - \chi_{B_{\mathbb{R}^P}(0,\beta)}(z), \\ 303 \end{aligned}$$

304 with A defined in (12).

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

$$306 \quad (20) \quad A^* = A_1^* + A_2^*,$$

307 with the pixel-wise multiplication:

$$308 \quad (21) \quad (\nabla v_1 - \nabla v_2)^* = \nabla v_1 - \nabla v_2,$$

309 and with:

$$310 \quad (22) \quad ((v_1 - v_2) \otimes \nabla)^* = \text{div}(I_2 \otimes (v_1 - v_2)^T),$$

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

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

$$\begin{aligned} 314 & \\ 315 \quad (23) \quad & \min_{u^{(t)}, w^{(t)}} \max_{p, z} \langle p(x) | \nabla u \rangle + \langle Aw | z \rangle + \langle \nabla v_2 | z \rangle \\ 316 & \quad + \lambda \int_{\Omega} w \|u - c_1\|_2^2 + (1 - w) \|u - c_2\|_2^2 \\ 317 & \quad - \chi_{B(0,\alpha)}(p) - \chi_{B_{\mathbb{R}^P}(0,\beta)}(z) + \chi_{\mathcal{R}}(u) + \chi_{[0,1]}(w). \end{aligned}$$

319 **2.4.3. Final reading of the algorithm.** Since the terms of Equation (23) are
 320 pixel-wise separable, we then remove the $N \times M$ notations. We apply Algorithm 1
 321 to (23) with the following identifications:

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

327 The proximal operators are given by the following lemmas.

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

$$329 \quad (24) \quad \text{prox}_{\sigma_w J^*} = P_{B_{\mathbb{R}^{N \times 4}}(0, \beta)}(\tilde{z} + \sigma_w \nabla v_2),$$

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

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

$$332 \quad (25) \quad \text{prox}_{-\sigma_w \langle \nabla v_2 | \cdot \rangle + \chi_{B_{\mathbb{R}^{N \times 4}}(0, \beta)}}(\tilde{z})$$

$$334 \quad = \underset{z \in \mathbb{R}^4}{\text{argmin}} \frac{\|z - \tilde{z}\|_2^2}{2\sigma_w} - \langle \nabla v_2 | z \rangle + \chi_{B_{\mathbb{R}^{N \times 4}}(0, \beta)}(z),$$

$$335$$

336 thus:

$$337 \quad (26) \quad \text{prox}_{-\sigma_w \langle \nabla v_2 | \cdot \rangle + \chi_{B_{\mathbb{R}^{N \times 4}}(0, \beta)}}(\tilde{z}) = P_{B_{\mathbb{R}^{N \times 4}}(0, \beta)}(\tilde{z} + \sigma_w \nabla v_2). \quad \square$$

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

$$339 \quad (27) \quad \text{prox}_{\tau_w(h(u, w) + H(w))}(\tilde{w}) = P_{[0, 1]}(\tilde{w} - \tau_w \lambda (\|u - c_1\|_2^2 - \|u - c_2\|_2^2)),$$

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

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

$$342 \quad \text{prox}_{\tau_w(h(u, w) + H(w))}(\tilde{w}) = \underset{w}{\text{argmin}} \frac{\|w - \tilde{w}\|_2^2}{2}$$

$$343 \quad + \tau_w \lambda w \|u - c_1\|_2^2 + \tau_w \lambda (1 - w) \|u - c_2\|_2^2$$

$$344 \quad + \chi_{[0, 1]}(w)$$

$$345 \quad = \underset{w}{\text{argmin}} \frac{\|w - \tilde{w}\|_2^2}{2}$$

$$346 \quad + \tau_w \lambda w (\|u - c_1\|_2^2 - \|u - c_2\|_2^2)$$

$$347 \quad + \chi_{[0, 1]}(w)$$

$$348 \quad (28) \quad = P_{[0, 1]}(\tilde{w} - \tau_w \lambda (\|u - c_1\|_2^2 - \|u - c_2\|_2^2)). \quad \square$$

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

$$351 \quad (29) \quad \text{prox}_{\tau_u(h(u, w) + F(u))}(\tilde{u}) = P_{\mathcal{R}} \left(\frac{\tilde{u} - \tau_u \lambda 2(w c_1 + (1 - w) c_2)}{1 + 2\tau_u \lambda} \right),$$

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

353 *Proof.* The function is

$$\begin{aligned}
354 \quad & \tau_u \lambda w \|u - c_1\|_2^2 + \tau_u \lambda (1 - w) \|u - c_2\|_2^2 + \chi_{\mathcal{R}}(u): \\
355 \quad & \text{prox}_{\tau_u(h(u,w)+F(u))}(\tilde{u}) = \operatorname{argmin}_u \|u - \tilde{u}\|_2^2 + 2\tau_u \lambda w \|u - c_1\|_2^2 \\
356 \quad & \quad \quad \quad + 2\tau_u \lambda (1 - w) \|u - c_2\|_2^2 + \chi_{\mathcal{R}}(u) \\
357 \quad & = \operatorname{argmin}_u \|u\|_2^2 - 2\langle u | \tilde{u} \rangle - 4\tau_u \lambda \langle u | w c_1 \rangle + 2\tau_u \lambda w \|u\|_2^2 \\
358 \quad & \quad \quad \quad - 4\tau_u \lambda \langle u | (1 - w) c_2 \rangle + 2\tau_u \lambda (1 - w) \|u\|_2^2 + \chi_{\mathcal{R}}(u) \\
359 \quad & = \operatorname{argmin}_u (2 + 2\tau_u \lambda) \|u\|_2^2 - 4\tau_u \lambda \langle u | w c_1 + (1 - w) c_2 \rangle \\
360 \quad & \quad \quad \quad - 2\langle u | \tilde{u} \rangle + \chi_{\mathcal{R}}(u) \\
361 \quad & = \operatorname{argmin}_u \|u\|_2^2 - 2 \left\langle u \left| \frac{\tilde{u} + \tau_u \lambda 2(w c_1 + (1 - w) c_2)}{2 + 2\tau_u \lambda} \right. \right\rangle \\
362 \quad & \quad \quad \quad + \chi_{\mathcal{R}}(u) \\
363 \quad (30) \quad & = P_{\mathcal{R}} \left(\frac{\tilde{u} + \tau_u \lambda 2(w c_1 + (1 - w) c_2)}{1 + \tau_u \lambda} \right). \quad \square \\
364
\end{aligned}$$

365 The final algorithm, solving (1) for one frame reads as Algorithm 2. $P_{B(0,\alpha)}$ and
366 $P_{B_{\mathbb{R}^P}(0,\beta)}$ represent the projection onto the L^2 ball of radius α and β respectively.
367 $P_{[0,1]}$ is the truncation of the coordinates between 0 and 1. $P_{\mathcal{R}}$ is the projection onto
368 the standard range for chrominances.

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

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

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

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

390 3.1. Overview of the User-guided

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

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

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

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

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

3: $z^0 \leftarrow Aw^0$

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

5: $p^{n+1} \leftarrow P_{B(0,\alpha)}(p^n + \sigma_u \nabla \bar{u}^n)$

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

7:

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

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

9: $\bar{w}^{n+1} \leftarrow 2w^{n+1} - w^n$

10: $\bar{u}^{n+1} \leftarrow 2u^{n+1} - u^n$

 11: **end for**

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

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

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

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

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

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

407 **3.2. User-guided Correction Model.** Assume that a first result \tilde{c}_1 has been
 408 computed from Model (1), from time 1 to time n , and that the user adds S scribbles
 409 on the unsuitable result on the n -th frame, providing S candidates \tilde{c}_i , $i = 2 \dots S + 1$.
 410 One candidate is providing for all the pixels of the video sequence.

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

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

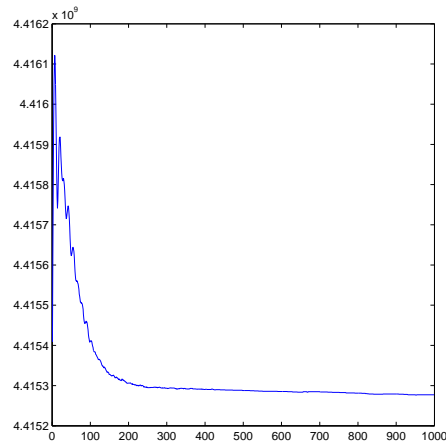


FIG. 5. The values of the functional decrease during the iterations of Algorithm 2 and become asymptotically constant. This behavior highlights the numerical convergence of this algorithm.

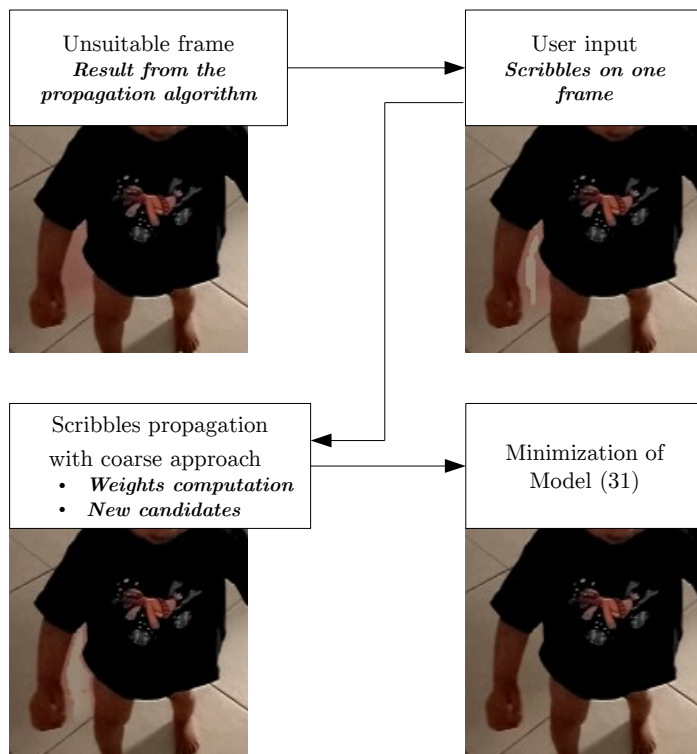


FIG. 6. Interactive Scribble-based Correction. The correction algorithm provides a trade-off between the colors produced by the frame-to-frame propagation method, and the ones given by the user. A coarse approach (the geodesic distance diffusion for instance) is used to provide a first scribble diffusion,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

417 where u is a 2D+t chrominance block:
418

$$\begin{aligned}
 419 \quad (31) \quad (\hat{u}, \hat{w}) = & \operatorname{argmin}_{u,w} \alpha \operatorname{TV}_{[t_1,n]}(u) \\
 & + \lambda \int_{\Omega \times [t_1,n]} \sum_{i=1}^{S+1} w_i^{(t)}(x) \|u^{(t)}(x) - \tilde{c}_i^{(t)}(x)\|_2^2 dx dt \\
 420 & \\
 421 & \\
 422 & + \chi_{\mathcal{R}}(u) + \chi_{\mathcal{E}}(w).
 \end{aligned}$$

423 The minimizer of the data-fidelity term

$$424 \quad (32) \quad \int_{\Omega \times [t_1,n]} \sum_{i=1}^{S+1} w_i^{(t)} \|u^{(t)}(x) - \tilde{c}_i^{(t)}(x)\|_2^2 dx dt,$$

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

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

$$429 \quad (33) \quad \operatorname{TV}_{[t_1,n]}(u) = \int_{\Omega \times [t_1,n]} (\gamma \|\Lambda \nabla Y\|_2^2 + \|\Lambda \nabla U\|_2^2 + \|\Lambda \nabla V\|_2^2)^{\frac{1}{2}},$$

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

$$431 \quad (34) \quad \Lambda := \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \mu \end{pmatrix}.$$

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

438 **3.3. Algorithm.** For the minimization of Model (31), the algorithm presented
439 in [27] is directly used. The composition of operator Λ with the gradient and the
440 divergence is not restrictive to apply the general algorithm of [27]. Finally, the mini-
441 mization of Model (31) is achieved with Algorithm 3, where c_*^n states for the closest
442 candidate \tilde{c}_i from u^n , and τ, σ are time steps. This Algorithm would contain an
443 update for the variable w with the following form:

$$444 \quad (35) \quad w^{n+1} \leftarrow \operatorname{argmin}_w \|w - w^n\|_2^2 + \rho \sum_{i=1}^{S+1} w_i \|u^n - \tilde{c}_i\|_2^2 + \chi_{\Delta}(w).$$

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

$$447 \quad (36) \quad w^{n+1} \leftarrow \operatorname{argmin}_w \sum_{i=1}^{S+1} w_i \|u^n - \tilde{c}_i\|_2^2 + \chi_{\Delta}(w),$$

448 and it is minimised by $w^* = (0, \dots, 1, 0, \dots, 0)$ with 1 in the i -th position, where
449 $\|u^n - c_i\|$ is the lowest one. Thus $\sum_{i=1}^{S+1} w^* c_i = c_j$ with c_i the closest candidate from
450 u^n and denoted by c_*^n .

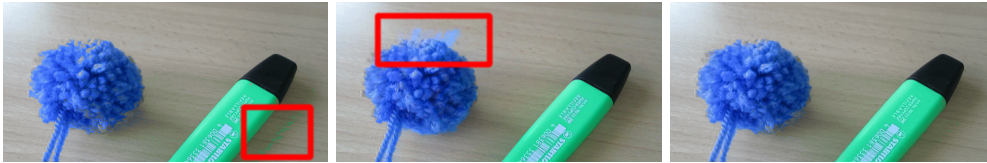
Algorithm 3 Minimization of (31).

```

1:  $u^0 = \sum_{i=1}^{S+1} w_i c_i$ .
2:  $p^0 \leftarrow \nabla u$ 
3: for  $n \geq 0$  do
4:    $p^{n+1} \leftarrow P_{B(0,\alpha)}(p^n + \sigma \Lambda \nabla \bar{u}^n)$ 
5:    $u^{n+1} \leftarrow P_{\mathcal{R}}\left(\frac{u^n + \tau (\operatorname{div}(\Lambda p^{n+1}) + 2\lambda c_*^n)}{1 + 2\tau_u \lambda}\right)$ 
6:    $\bar{u}^{n+1} \leftarrow 2u^{n+1} - u^n$ 
7: end for

```

451 With the combination of Algorithm 2 for color propagation and Algorithm 3 for
452 scribble correction, we propose, in this work, a complete framework for video coloriza-
453 tion. It is composed of a frame-to-frame propagation approach, which is unsupervised,
454 and a correction step when occlusions or dis-occlusions occur. The interaction between
455 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).*

456 **4. Implementation Details.** In this section, we describe implementation de-
457 tails. In particular, we present the computation of the correspondence maps and the
458 initialisation of the correction algorithm. We also explain how the two models for
459 propagation and correction interact with each other.

4.1. Global Workflow of the Proposed Video Colorization Method.

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

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

473 propagation efficiency reduces user’s interventions. This efficiency is demonstrated in
474 Section 5.

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

481 **4.2. Computation of the Correspondence Maps.** Our method propagates
482 colors from an initial frame to the final one, through an optimal correspondence map
483 that is computed from initial correspondence maps with Equation (1). In this work,
484 we compute two initial maps, from the luminance channel of the video sequence, with
485 the PatchMatch method [3] and the TVL1 optical flow [7].

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

$$490 \quad (37) \quad \hat{y} = \operatorname{argmin}_{y \in \Omega} \|P(x) - P(y)\|_2,$$

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

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

$$499 \quad (38) \quad \rho(u, v) = \partial_t u + \langle \nabla_{x,y} u | v \rangle + \beta u,$$

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

$$502 \quad (39) \quad \min_v \operatorname{TV}(v) + \operatorname{TV}(u) + \|\rho(u, v)\|_1,$$

503 which gives the TVL1 optical flow.

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

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

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

521 **4.3. Scribble Correction with Geodesic Distance.** With the correction
 522 Model (31), we compute a solution from the frame-to-frame propagation result and
 523 user’s scribbles. This model performs a scribble diffusion thanks to the total variation
 524 minimization. To help the propagation of the scribbles, a technique inspired by [38]
 525 is used. The geodesic distance is computed on a three dimensional block, composed
 526 of the two spatial dimensions with the temporal one. This geodesic distance pro-
 527 vides, for each pixel, a weight w_i associated to the chrominance candidate \tilde{c}_i given
 528 by one of the scribbles. \tilde{c}_1 is the previous frame-to-frame propagation result, \tilde{c}_i with
 529 $i = 2, \dots, S + 1$ is the chrominance given by one scribble.

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

$$533 \quad (40) \quad (\theta + \|\nabla u\|_2^2)^{-r},$$

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

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

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

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

555 **4.4. Parameter Settings.** The results presented in Section 5 have been per-
 556 formed with the following set of parameters: $\lambda = 0.1$, $\alpha = 25$ and $\gamma = 35$, in Equa-
 557 tions (1) and (31), $\beta = 0.1$ in Equation (1), $\mu = 0.1$ in Equation (31), and $\theta = 45$,
 558 $r = 8$ in Equation (40). These parameters are the same for all the videos. We consider
 559 2000 iterations for Algorithms 2 and 3, that are computed in about 1 second per frame
 560 at image resolution 352×256 with an unoptimized NVIDIA CUDA implementation.
 561 This computation time is about the same for the propagation step and the correction
 562 one. It is fast enough to consider the user-guided interaction.

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

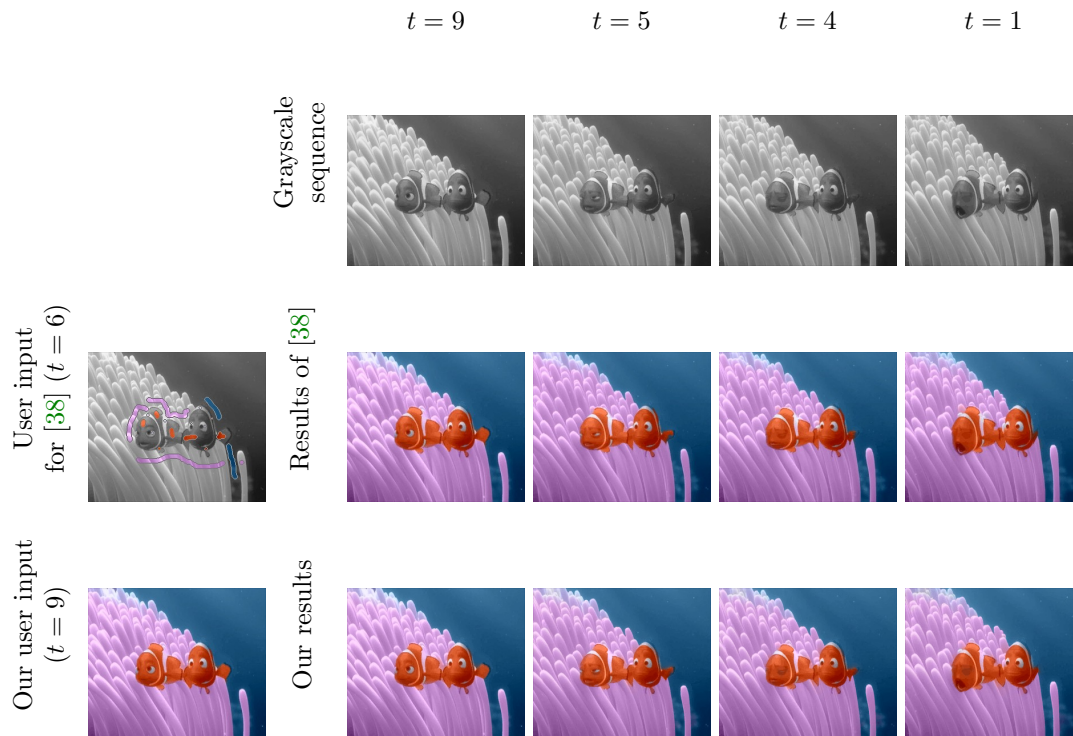


FIG. 8. Comparison with [38] on the Nemo video sequence. We use the last colorized frame given by the method of [38] and we propagate its colors. No scribble correction has been used for our results. With our method, the fishtail is better colorized, zooms are available in Figure 9.

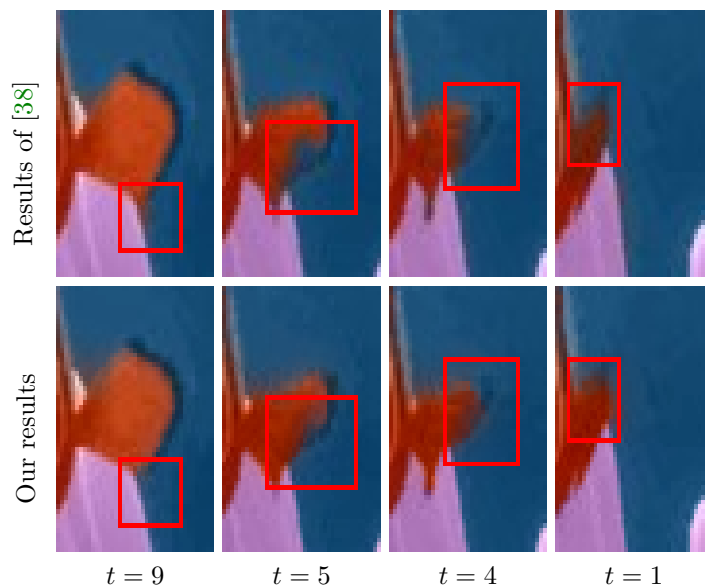


FIG. 9. Zooms on Figure 8. The contours are better preserved with our approach (see, e.g., contours in the red boxes).

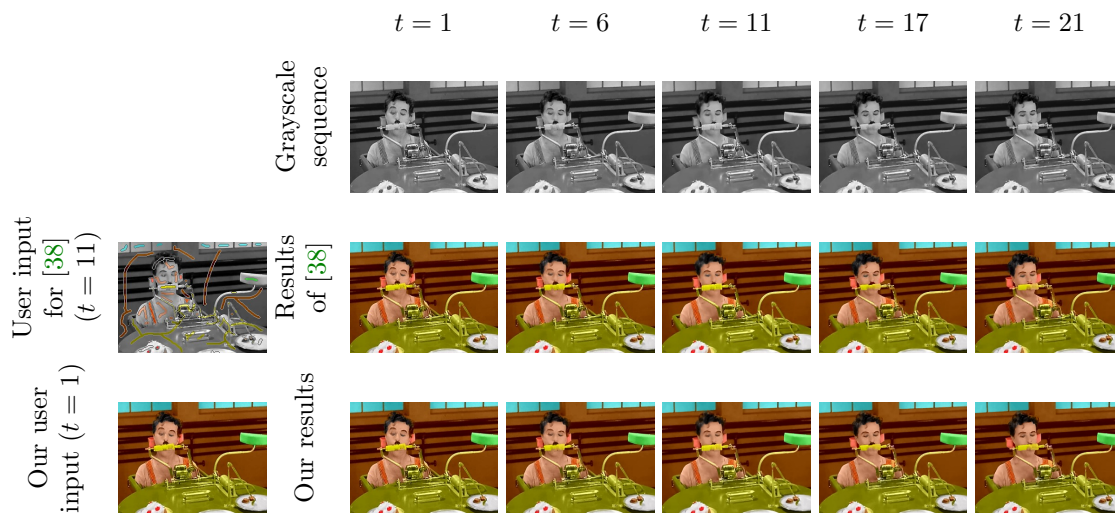
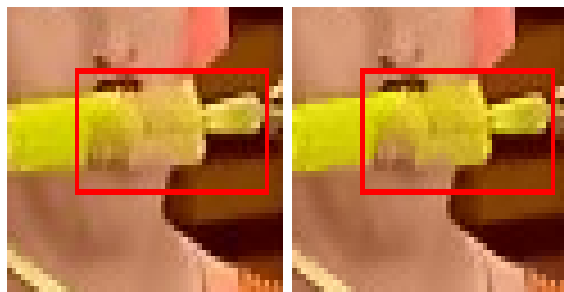


FIG. 10. Comparison with [38] on the Chaplin video sequence. We use the first colored frame given by the method of [38] and we propagate its colors. No scribble correction has been used for our results. With our method, the corncob is better colored, zooms are available in Figure 11.



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.

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

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

570 **5.1. Comparison with Chrominance Blending [38].** First, let us compare
 571 our method to [38]. The two videos tested in Figures 8 and 10 have been taken
 572 from [38]. The video sequences *Nemo* and *Chaplin* are colorized by a propagation of
 573 one frame. The initial colored frame has been taken from the results of [38].

574 Figure 8 (resp. Figure 10) shows frames of the results from [38] and the colori-
 575 zation performed with our propagation algorithm on the video *Nemo* (resp. on the
 576 video *Chaplin*).

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

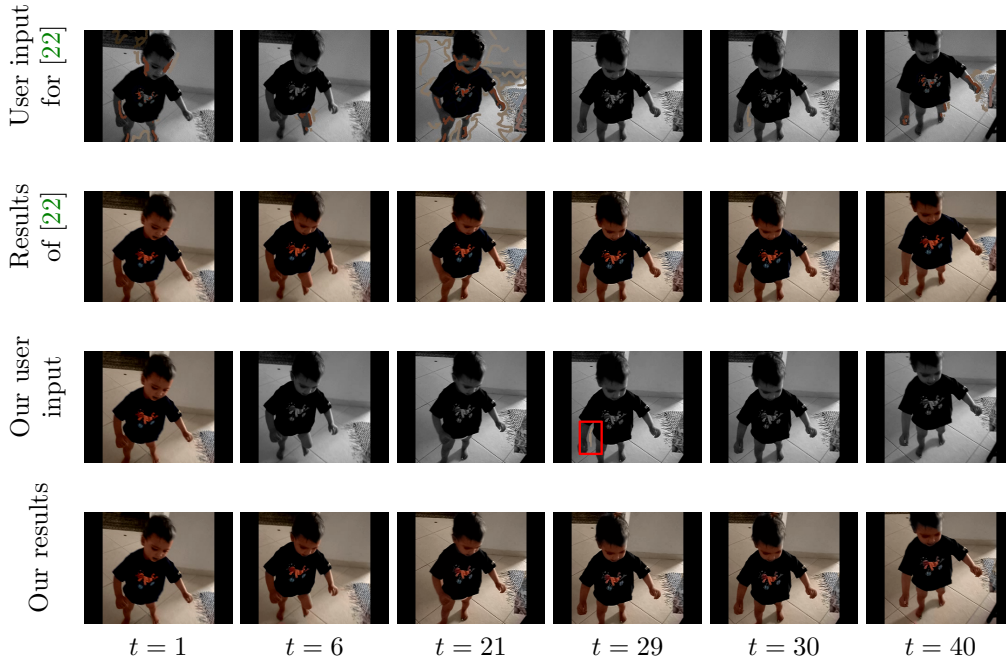


FIG. 12. Comparison with [22] on the Toddler video sequence. We use the first colorized frame given by the method of [22] and we propagate its colors. Our method requires one scribble at time $t = 29$ (visible in the red box). Details of the correction step are available in Figure 6.

579 the same.

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

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

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

593 **5.2. Comparison with Quadratic Optimization [22].** Now, let us compare
 594 our method on examples taken from [22].

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

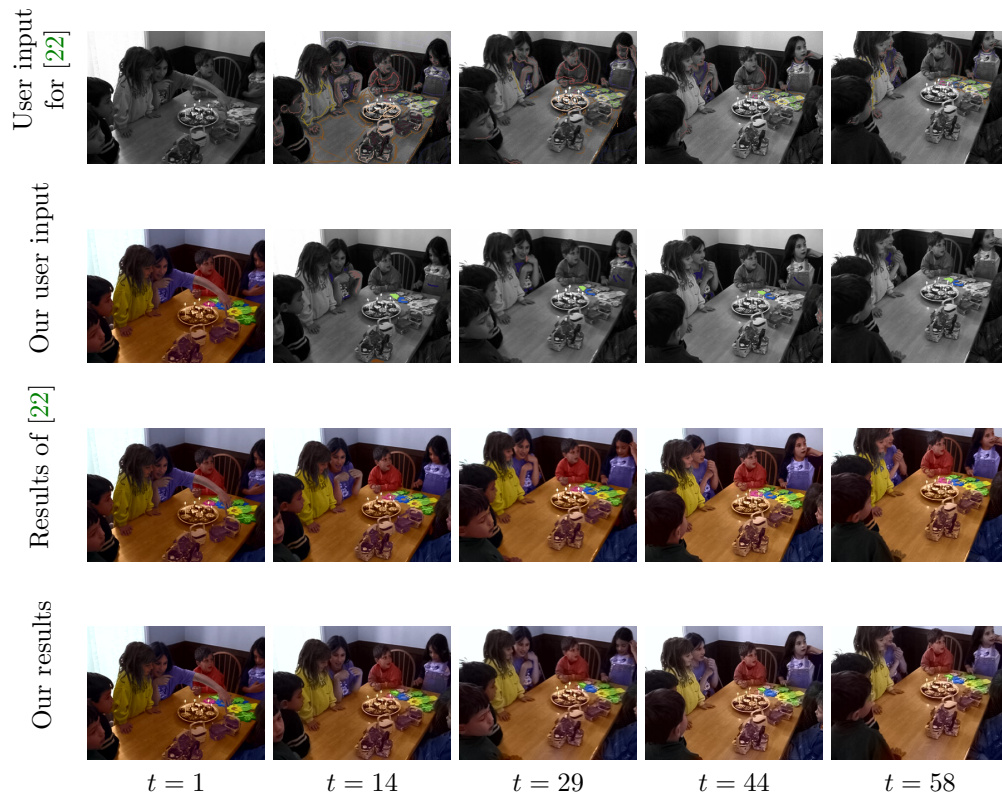


FIG. 13. Comparison of our method with [22] for the Birthday video sequence. We use the first colored frame given by the method of [22] and we propagate its colors. For our method, the user draws about 50 scribbles to correct the propagation algorithm.

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

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

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

616 **5.3. Experiments on Historical Videos.** In this section, we consider the
 617 challenging problem of colorization of historical videos. The difficulties come from
 618 the noise, the flickering effect and the scratches.

619 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 $t = 25$.

620 frame is colorized with [29]. In the *De Gaulle* video sequence, the thumb is hidden at
 621 the beginning of the sequence before appearing. Thus, it is unsuitably colorized by the
 622 propagation algorithm. We add a scribble onto the thumb when it appears (in the red
 623 box) and the correction algorithm integrates it. Finally, the propagation algorithm
 624 computes the colorization of the sequence. The 50 frames with image resolution of
 625 352×256 pixels are colorized in about 1 min with our GPU implementation.

626 Finally, we apply our algorithm to the sequence *Pieds Nickelés* that is a very
 627 noisy video (Figure 15). This sequence is processed with the method of [12] to remove
 628 flicker effect. It contains a residual flicker effect as well as many scratches. Moreover,
 629 some of the displacements are very large, for instance, the heads of the characters. In
 630 spite of these challenging problems, our method is able to propagate colors over the
 631 100 frames of the whole sequence without any correction. The grayscale sequence of
 632 our method is available on the first and the third rows, whereas the result is on the
 633 second and fourth ones. The initialization is the initial frame at $t = 1$.

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

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

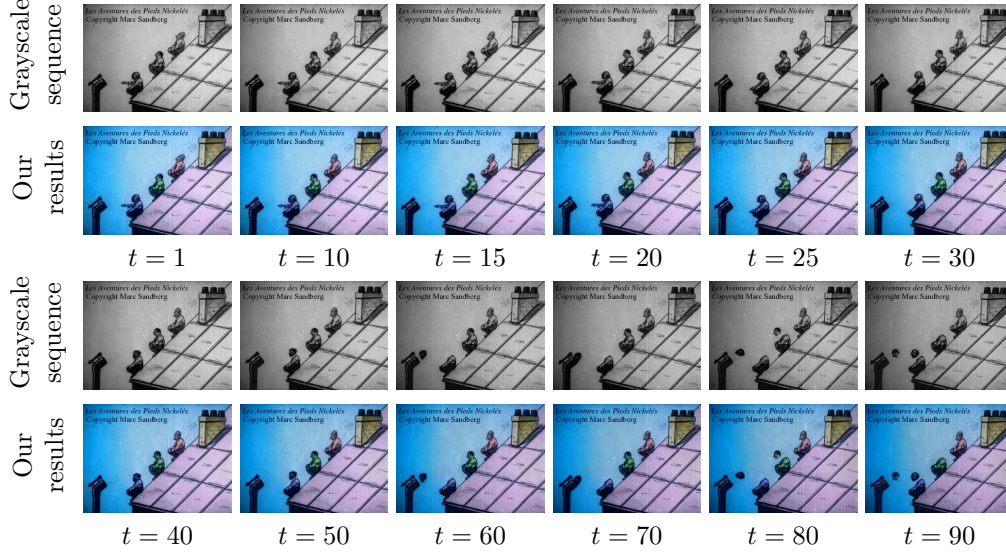


FIG. 15. Result of our method for the Pieds Nickelés [12] video sequence. The initialization is the first frame at time $t = 1$, colorized with the method of [28]. For this sequence, besides the colorization of the first frame, no scribble has been added. This experiment shows that our approach is reliable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

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

645 Appendix A. Computation of the Map Operator.

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

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

$$649 \quad (41) \quad \|A\| \leq \|A_1\| \|\nabla\| + \|A_2\|.$$

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

652 Let us compute $\|A_1\|$:

$$654 \quad (42) \quad \|A_1\| = \|A_1^*\| = \left\| \begin{pmatrix} \varphi_1 - \varphi_2 & 0 & \psi_1 - \psi_2 & 0 \\ 0 & \varphi_1 - \varphi_2 & 0 & \psi_1 - \psi_2 \end{pmatrix} \right\|$$

$$655 \quad \leq \left\| \begin{pmatrix} \varphi_1 - \varphi_2 & 0 \\ 0 & \varphi_1 - \varphi_2 \end{pmatrix} \right\| + \left\| \begin{pmatrix} \psi_1 - \psi_2 & 0 \\ 0 & \psi_1 - \psi_2 \end{pmatrix} \right\|$$

$$656 \quad = |\varphi_1 - \varphi_2| + |\psi_1 - \psi_2|.$$

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

$$660 \quad (43) \quad \|A_1\| \leq 2(N + M).$$

661 Let us compute $\|A_2\|$:

662

$$\begin{aligned}
 663 \quad (44) \quad \|A_2\| &= \|A_2^*\| = \left\| \begin{pmatrix} \partial_x \varphi_1 - \partial_x \varphi_2 & \partial_y \varphi_1 - \partial_y \varphi_2 \\ \partial_x \psi_1 - \partial_x \psi_2 & \partial_y \psi_1 - \partial_y \psi_2 \end{pmatrix} \right\| \\
 664 &\leq |\partial_x \varphi_1 - \partial_x \varphi_2| + |\partial_y \varphi_1 - \partial_y \varphi_2| \\
 665 &\quad + |\partial_x \psi_1 - \partial_x \psi_2| + |\partial_y \psi_1 - \partial_y \psi_2|. \\
 666 & \\
 667 &
 \end{aligned}$$

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

$$670 \quad (45) \quad \|A_2\| \leq 4(N + M),$$

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

672 Finally, a bounding of A reads as:

$$673 \quad (46) \quad \|A\| \leq 2\sqrt{8}(N + M) + 4(N + M).$$

674 The operator norm depends linearly on the size of the frame. \square

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

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

$$678 \quad (47) \quad \min_{u \in \mathbb{R}^U, w \in \mathbb{R}^W} F(u) + h(u, w) + H(w)$$

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

$$683 \quad (48) \quad \begin{cases} \tilde{\sigma}_n < \frac{\gamma_1}{L_1(w^n)} \\ \tilde{\tau}_n < \frac{\gamma_2}{L_2(u^{n+1})}, \end{cases}$$

684 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
 685 the Lipschitz constant of $\nabla_w h(u^{n+1}, w)$.

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

- 687 • $F(u) = \alpha \text{TV}_{\mathcal{E}}(u) + \chi_{\mathcal{R}}(u)$,
- 688 • $h(u, w) = \frac{\lambda}{2} \int_{\Omega} \sum_{i=1}^{\mathcal{M}} w_i(x) \|u(x) - c_i(x)\|_2^2 dx$,
- 689 • $H(w) = \beta \text{TV}(v^{(t)}) + \chi_{\Delta}(w^{(t)})$.

690 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].

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

691 The proximal operators of H and G can be computed with the iterative primal-
 692 dual algorithm of Chambolle and Pock [7]. Indeed, let us compute $\text{prox}_{\tilde{\sigma}_n F}(\tilde{u})$ in
 693 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].

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

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

1: **for** $n \geq 0$ **do**
2:

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

3:

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

4:
5: $\bar{w}^{n+1} \leftarrow 2w^{n+1} - w^n$
6: **end for**

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

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

$$698 \quad (49) \quad \nabla_u h(u, w) = \nabla_u \left(\frac{\lambda}{2} \int_{\Omega} \sum_{i=1}^{\mathcal{M}} w_i \|u - c_i\| dx \right)$$

$$699 \quad (50) \quad = \lambda \left(u - \sum_{i=1}^{\mathcal{M}} w_i c_i \right).$$

700
701

$$702 \quad (51) \quad \nabla_w h(u, w) = \nabla_w \left(\frac{\lambda}{2} \int_{\Omega} \sum_{i=1}^{\mathcal{M}} w_i \|u - c_i\| dx \right)$$

$$703 \quad (52) \quad = \lambda (\|u - c_i\|)_i.$$

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

707 **PROPOSITION B.1.** *Let us consider Algorithm 7 with only one iteration for the*
708 *inner loops (i.e., Algorithm 5 and 6) and a common value for $\tilde{\sigma}$ and $\tilde{\tau}$. Let us choose*

709 $\tau_u = \frac{\tilde{\tau}_u}{1 + \tilde{\tau}_u \tilde{\sigma} - \tilde{\tau}_u \lambda \tilde{\sigma}^2}$, $\sigma_u = \tilde{\sigma}_u$, $\tau_w = \frac{\tilde{\tau}_w}{1 + \tilde{\tau}_w \tilde{\tau}}$ and $\sigma_w = \tilde{\sigma}_w$. *Then, Algorithm 7 is*
710 *the same as Algorithm 2.*

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

```

1: for  $k \geq 0$  do
2:    $\tilde{w} \leftarrow w^n - \tilde{\tau}\lambda(\|u^n - c_i\|)_i$ 
3:   for  $n \geq 0$  do
4:      $z^{n+1} \leftarrow P_{B_{\mathbb{R}^F}(0,\beta)}(z^n + \tilde{\sigma}_w((v_1 - v_2) \otimes \nabla \bar{w}^n + (\nabla v_1 - \nabla v_2)\bar{w}^n + \tilde{\sigma}_w \nabla v_2))$ 
5:
6:      $w^{n+1} \leftarrow P_{[0,1]} \left( \frac{1}{1 + \tilde{\tau}_w \tilde{\tau}} [w^n + \tilde{\tau}_w((\nabla v_1 - \nabla v_2)z^{n+1} \right.$ 
7:
8:          $\left. + \operatorname{div}(I_2 \otimes (v_1 - v_2)^T v^{n+1}) + \tilde{\tau}_w \tilde{\tau} \tilde{w}) \right]$ 
9:
10:    6:    $\bar{w}^{n+1} \leftarrow 2w^{n+1} - w^n$ 
11:    7:   end for
12:    8:    $\tilde{u} \leftarrow u^n - \tilde{\sigma}\lambda \left( u^n - \sum_{i=1}^{\mathcal{M}} w_i^{n+1} c_i \right)$ 
13:    9:   for  $n \geq 0$  do
14:    10:   $p^{n+1} \leftarrow P_{B(0,\alpha)}(p^n + \tilde{\sigma}_u \nabla \bar{u}^n)$ 
15:    11:   $u^{n+1} \leftarrow P_{\mathcal{R}} \left( \frac{u^n + \tilde{\tau}_u (\operatorname{div}(p^{n+1}) + \tilde{\sigma} \tilde{u})}{1 + \tilde{\tau}_u \tilde{\sigma}} \right)$ 
16:    12:   $\bar{u}^{n+1} \leftarrow 2u^{n+1} - u^n$ 
17:    13:  end for
18: 14: end for

```

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

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

$$717 \quad (53) \quad u^{n+1} \leftarrow P_{\mathcal{R}} \left(\frac{u^n + \tilde{\tau}_u \left(\operatorname{div}(p^{n+1}) + \tilde{\sigma} \left(u^n - \tilde{\sigma}\lambda \left(u^n - \sum_{i=1}^{\mathcal{M}} w_i^n c_i \right) \right) \right)}{1 + \tilde{\tau}_u \tilde{\sigma}} \right).$$

718 with simplification:

$$719 \quad (54) \quad u^{n+1} \leftarrow P_{\mathcal{R}} \left(\frac{u^n(1 + \tilde{\tau}_u \tilde{\sigma} - \tilde{\tau}_u \lambda \tilde{\sigma}^2) + \tilde{\tau}_u \left(\operatorname{div}(p^{n+1}) + \lambda \tilde{\sigma}^2 \sum_{i=1}^{\mathcal{M}} w_i^n c_i \right)}{1 + \tilde{\tau}_u \tilde{\sigma}} \right).$$

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

$$721 \quad (55) \quad \frac{1 + \tilde{\tau}_u \tilde{\sigma}}{1 + \tilde{\tau}_u \tilde{\sigma} - \tilde{\tau}_u \lambda \tilde{\sigma}^2} = 1 + \tilde{\lambda} \tau_u.$$

722 Thus Equation (54) becomes:

$$723 \quad (56) \quad u^{n+1} \leftarrow P_{\mathcal{R}} \left(\frac{u^n + \tau_u \left(\operatorname{div}(p^{n+1}) + \lambda \sum_{i=1}^{\mathcal{M}} w_i^n c_i \right)}{1 + \lambda \tau_u} \right),$$

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

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

$$727 \quad (57) \quad w^{n+1} \leftarrow P_{[0,1]} \left(\frac{1}{1 + \tilde{\tau}_w \tilde{\tau}} [w^n + \tilde{\tau}_w ((\nabla v_1 - \nabla v_2) z^{n+1} \right. \\ 728 \quad \left. + \operatorname{div}(I_2 \otimes (v_1 - v_2)^T v^{n+1}) + \tilde{\tau}_w \tilde{\tau} (w^n - \tilde{\tau} \lambda (\|u^n - c_i\|)_i)] \right).$$

730 With simplification:

$$732 \quad (58) \quad w^{n+1} \leftarrow P_{[0,1]} \left(w^n + \frac{\tilde{\tau}_w}{1 + \tilde{\tau}_w \tilde{\tau}} ((\nabla v_1 - \nabla v_2) z^{n+1} \right. \\ 733 \quad \left. + \operatorname{div}(I_2 \otimes (v_1 - v_2)^T v^{n+1}) - \frac{\tilde{\tau}_w \tilde{\tau}^2}{1 + \tilde{\tau}_w \tilde{\tau}} \lambda (\|u^n - c_i\|)_i \right)$$

735 Choosing $\tau_w = \frac{\tilde{\tau}_w}{1 + \tilde{\tau}_w \tilde{\tau}}$, and $\tilde{\lambda} = \lambda \tilde{\tau}^2$, thus Equation (58) is the same computation
736 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
737 that $\tilde{\tau} = \tilde{\sigma}$. That is not a restrictive hypothesis, since it is possible to use the same
738 time step in the PALM algorithm, by choosing the minimum of them as a common
739 value to ensure the convergence. \square

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

749 **Acknowledgment.** This study has been carried out with financial support from
750 the French State, managed by the French National Research Agency (ANR) in the
751 frame of the Investments for the future Programme IdEx Bordeaux (ANR-10-IDEX-
752 03-02). J.-F. Aujol is a member of Institut Universitaire de France.

753

REFERENCES

- 754 [1] P. ARIAS, G. FACCIOLO, V. CASELLES, AND G. SAPIRO, *A variational framework for exemplar-*
755 *based image inpainting*, International Journal of Computer Vision, 93 (2011), pp. 319–347.
756 [2] J.-F. AUJOL, S. LADJAL, AND S. MASNOU, *Exemplar-based inpainting from a variational point*
757 *of view*, SIAM Journal on Mathematical Analysis, 42 (2010), pp. 1246–1285.
758 [3] C. BARNES, E. SHECHTMAN, A. FINKELSTEIN, AND D. GOLDMAN, *Patchmatch: a randomized*
759 *correspondence algorithm for structural image editing*, in Transactions on Graphics, ACM,
760 2009, pp. 24–32.
761 [4] M. J. BLACK AND P. ANANDAN, *The robust estimation of multiple motions: Parametric and*
762 *piecewise-smooth flow fields*, Computer vision and image understanding, 63 (1996), pp. 75–
763 104.

- 764 [5] J. BOLTE, S. SABACH, AND M. TEBoulLE, *Proximal alternating linearized minimization for*
765 *nonconvex and nonsmooth problems*, *Mathematical Programming*, 146 (2014), pp. 459–
766 494.
- 767 [6] A. CHAMBOLLE, *An algorithm for total variation minimization and applications*, *Journal of*
768 *Mathematical Imaging and Vision*, 20 (2004), pp. 89–97.
- 769 [7] A. CHAMBOLLE AND T. POCK, *A first-order primal-dual algorithm for convex problems with*
770 *applications to imaging*, *Journal of Mathematical Imaging and Vision*, 40 (2011), pp. 120–
771 145.
- 772 [8] T. F. CHAN AND L. A. VESE, *Active contours without edges*, *IEEE Transactions on Image*
773 *Processing*, 10 (2001), pp. 266–277.
- 774 [9] G. CHARPIAT, M. HOFMANN, AND B. SCHÖLKOPF, *Automatic image colorization via multimodal*
775 *predictions*, in *European Conference on Computer Vision*, Springer, 2008, pp. 126–139.
- 776 [10] P. L. COMBETTES AND V. R. WAJS, *Signal recovery by proximal forward-backward splitting*,
777 *Multiscale Modeling & Simulation*, 4 (2005), pp. 1168–1200.
- 778 [11] S. CONTI, J. GINSTER, AND M. RUMPF, *A BV functional and its relaxation for joint motion*
779 *estimation and image sequence recovery*, *ESAIM: Mathematical Modelling and Numerical*
780 *Analysis*, 49 (2015), pp. 1463–1487.
- 781 [12] J. DELON AND A. DESOLNEUX, *Stabilization of flicker-like effects in image sequences through*
782 *local contrast correction*, *SIAM Journal on Imaging Sciences*, 3 (2010), pp. 703–734.
- 783 [13] D. FORTUN, P. BOUTHEMY, AND C. KERVRANN, *Aggregation of local parametric candidates*
784 *with exemplar-based occlusion handling for optical flow*, *Computer Vision and Image Un-*
785 *derstanding*, (2015).
- 786 [14] M. GHANBARI, *The cross-search algorithm for motion estimation*, *IEEE Transactions on Com-*
787 *munications*, 38 (1990), pp. 950–953.
- 788 [15] R. K. GUPTA, A. Y.-S. CHIA, D. RAJAN, E. S. NG, AND H. ZHIYONG, *Image colorization using*
789 *similar images*, in *ACM International Conference on Multimedia*, 2012, pp. 369–378.
- 790 [16] J.-H. HEU, D.-Y. HYUN, C.-S. KIM, AND S.-U. LEE, *Image and video colorization based on*
791 *prioritized source propagation*, in *IEEE International Conference on Image Processing*,
792 2009, pp. 465–468.
- 793 [17] D.-Y. HYUN, J.-H. HEU, C.-S. KIM, AND S.-U. LEE, *Prioritized image and video colorization*
794 *based on gaussian pyramid of gradient images*, *Journal of Electronic Imaging*, 21 (2012),
795 pp. 023027–1.
- 796 [18] R. IRONY, D. COHEN-OR, AND D. LISCHINSKI, *Colorization by example*, in *Eurographics Symp.*
797 *on Rendering*, vol. 2, Citeseer, 2005.
- 798 [19] V. G. JACOB AND S. GUPTA, *Colorization of grayscale images and videos using a semiautomatic*
799 *approach*, in *IEEE International Conference on Image Processing*, 2009, pp. 1653–1656.
- 800 [20] M. LANG, O. WANG, T. AYDIN, A. SMOLIC, AND M. GROSS, *Practical temporal consistency for*
801 *image-based graphics applications*, in *Transactions on Graphics*, ACM, July 2012, pp. 34–
802 42.
- 803 [21] C. LANNAUD, *Fallait-il coloriser la guerre ?*, *L'express*, (2009).
804 http://www.lexpress.fr/culture/tele/fallait-il-coloriser-la-guerre_789380.html.
- 805 [22] A. LEVIN, D. LISCHINSKI, AND Y. WEISS, *Colorization using optimization*, in *Transactions on*
806 *Graphics*, ACM, 2004, pp. 689–694.
- 807 [23] B. D. LUCAS, T. KANADE, ET AL., *An iterative image registration technique with an application*
808 *to stereo vision.*, in *International Joint Conference on Artificial Intelligence*, vol. 81, 1981,
809 pp. 674–679.
- 810 [24] B. S. MANJUNATH AND W.-Y. MA, *Texture features for browsing and retrieval of image data*,
811 *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18 (1996), pp. 837–842.
- 812 [25] Z. PAN, Z. DONG, AND M. ZHANG, *A new algorithm for adding color to video or animation*
813 *clips*, UNION Agency-Science Press, 2004.
- 814 [26] G. PEYRÉ, *Toolbox fast marching - a toolbox for fast marching and level sets computations*,
815 2008. <http://www.mathworks.com>.
- 816 [27] F. PIERRE, J.-F. AUJOL, A. BUGEAU, N. PAPADAKIS, AND V.-T. TA, *Luminance-chrominance*
817 *model for image colorization*, *SIAM Journal on Imaging Sciences*, 8 (2015), pp. 536–563.
- 818 [28] F. PIERRE, J.-F. AUJOL, A. BUGEAU, AND V.-T. TA, *A unified model for image colorization*,
819 *in Color and Photometry in Computer Vision (ECCV Workshop)*, 2014, pp. 1–12.
- 820 [29] F. PIERRE, J.-F. AUJOL, A. BUGEAU, AND V.-T. TA, *Colociel*. Dépôt Agence
821 de Protection des Programmes N° IDDN.FR.001.080021.000.S.P.2016.000.2100, 2016.
822 <http://www.labri.fr/perso/fpierre/colociel.zip>.
- 823 [30] M. H. QUANG, S. H. KANG, AND T. M. LE, *Image and video colorization using vector-valued*
824 *reproducing kernel hilbert spaces*, *Journal of Mathematical Imaging and Vision*, 37 (2010),
825 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. ŠŮKORA, 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.