SIViP (2014) 8:1681–1689
DOI 10.1007/s11760-012-0402-5

ORIGINAL PAPER

Colorization using edge-preserving smoothing filter

Ahmed A. Hussein · Xiaochun Yang

Received: 2 November 2011 / Revised: 12 November 2012 / Accepted: 15 November 2012 / Published online: 4 December 2012

Abstract Colorization is the process of replacing a scalar value stored at each pixel of a grayscale image or film by a vector in a multidimensional color space. Mapping between scalar and color is therefore not unique, and colorization is ambiguous in nature and requires some amount of human interaction or external information. This paper presents a new method of interactive colorization by introducing a new concept to the bilateral filter. We adopt the bilateral filter as a range weights filter (bilateral filter without Gaussian spatial domain), and we show that the Nikolaou filter is equivalent to the range weights bilateral filter, but has better behavior near the edges. In our algorithm, the user selects grayscale image regions by directly painting these regions; the user does not need to paint over the whole object. Each selection can be automatically expanded from the user's paint brush and aligned with the object boundary. Robustness and quality of the results obtained over a collection of several challenging images demonstrate the efficiency of this new method for some difficult cases, such as human faces or images with confusing lighting.

Keywords Colorization · Bilateral filter · Nikolaou filter · Pixels interpolation

A. A. Hussein (✉) · X. Yang
School of Information Science and Engineering, Northeastern University, #3-11 Wenhua Rd., Heping District, Shenyang 110819, Liaoning Province, People's Republic of China
e-mail: ah_h72@yahoo.com

X. Yang
e-mail: yangxc@mail.neu.edu.cn

List of symbols

I	Digital image
s	The spatial locations of pixels at s
p	The spatial locations of pixels at p
I_s	Image at position s
I_p	Image at position p
W_s	Normalization at position s
Ω	Set of image pixels
$\sum_{p\in\Omega}$	Sum over all pixels p
f	The spatial domain
g	The range domain
σ_s	Standard deviation of the spatial similarity
σ_r	Standard deviation of the range (intensity/color) similarity
$\lvert.\rvert$	Absolute value
d_i	Intensity differences
k	Parameter (controls the amount of colors that will diffuse across edges)
t	Discrete time (iterations)
Y	Luminance of the pixels in a grayscale image
Y_p	Luminance of the pixels at position p
Y_s	Luminance of the pixels at position s
U, V	Chrominance (color information)
U_p, V_p	Color hints

1 Introduction

Colorization is the art of adding color to a monochrome image or movie. The idea of "coloring" photos and films is not new. Hand coloring of photographs is as old as photography itself [1].

The colorization problem amounts to replacing a scalar value stored at each pixel of a grayscale image

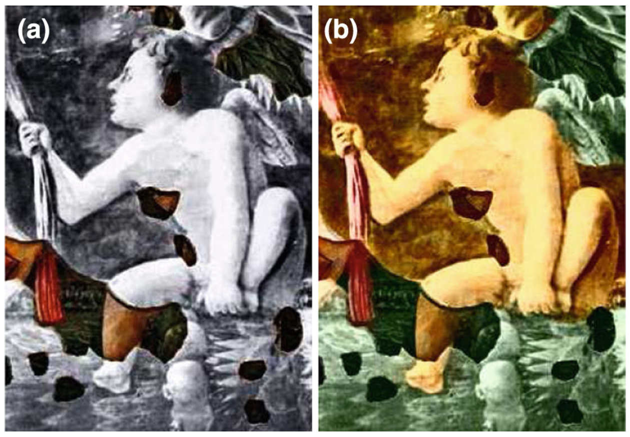

Fig. 1 A. Mantegna's fresco's recolorization. **a** A *grayscale* photo with a few *color pieces*. **b** The original color is reconstructed using our method

(e.g. luminance) by a vector in a multi-dimensional color space (e.g. a three-dimensional vector with luminance, saturation and hue). This is a very time-consuming and labor-intensive process and therefore is generally a severely under-constrained and ambiguous problem [2]. Due to these ambiguities, human interaction usually plays a large role in the colorization process.

This colorization problem was motivated by recovering frescoes' painted by A. Mantegna in an Italian church that was destroyed during World War II. There are photos of the full frescoes available in black and white, while only a few real pieces of frescoes with the original colors remain. The objective was to reconstruct the original color of the frescoes (image) using the few remaining real pieces of the original (with color) and the full black and white grayscale photos of the frescoes [3,4]. Figure 1 shows an example of A. Mantegna's fresco's recolorization using our method. A grayscale photo with a few color pieces(left), and our method extends the colors of the real pieces to reconstruct the original color of the photo (right).

This paper presents a new method of interactive colorization by introducing a new concept to the bilateral filter [5]. We adopt bilateral filter as a range weights filter (bilateral filter without Gaussian spatial domain), and we show that the Nikolaou filter [6] is equivalent to the range weights bilateral filter, but it is more robust near the edges.

Instead of removing the noise or unnecessary details from the images, in our algorithm, the user selects grayscale image regions by directly painting these regions with a paint brush. The user does not need to paint over the whole object. Instead, the selection can be automatically expanded from a user's paint brush and aligned with the object boundary by using a Nikolaou filter as a simple and effective Edge-Preserving Smoothing Filter(EPSF).

Robustness and quality of the results obtained over a collection of several challenging images demonstrate the efficiency of the new method for some difficult cases, such as human faces or images with confusing lighting.

1.1 Previous work

In our literature review, we divided the algorithms of colorization into three groups depending on the way those algorithms work:

In manual algorithms, the users segment the image into regions and colorize each region by hand. In Markle's original colorization process [7], a color mask is manually painted for at least one reference frame per shot. Motion detection and tracking are then applied. It appears that these systems still rely on defining regions and tracking them between the frames of a shot.

The second group of algorithms depends on transferring colors between images. The critical part of these algorithms is choosing the source image to match with the target image. For example, the process of selecting a source image representing the sea is not successful for coloring an image that represents the forest.

Welsh et al. [8] present a semi-automatic technique for colorizing a grayscale image by transferring color from reference data. The idea is to examine the luminance values in the neighborhood of each pixel in the target image and transfer the color from pixels with matching neighborhoods in the reference image.

Eisemann et al. [9] use the bilateral filter to correct the artifacts caused by the flash shadows, which takes an average of non-shadow flash-photo pixels, weighted by their similarity in the no-flash image and their pixel distance to the pixel to be corrected. They rely on using a shadow matte to transfer the color information from outside the shadow matte to fill in the missing information inside the shadow matte. This worked well in some cases, but failed in many, particularly where the kernel is large(over detail), which it blurs by nature, especially over softer edges. Furthermore, making shadow matting robust and practical remains a continuing challenge for many researchers [10–12].

Charpiat et al. [13] avoid using an automatic segmentation of the training or test images by using the probability distribution of all possible colors and also predicts the expected variation of color at each pixel. The algorithm then uses graph cuts to maximize the probability of the whole colored image at the global level.

Kekre et al. [14] propose a technique for generating a color palette using vector quantization codebook generation. Newly introduced Kekre's Biorthogonal color spaces are used, and then, the algorithm uses a color palette for colorizing grayscale images.

Pouli et al. [15] propose a color transfer technique that can progressively reshape the histogram of a given image to match it to the histogram of another. The approach relies on the novel idea of scale-space manipulation of the histograms, which allows the user to select how well the color palette of the input image should be matched to that of the target.

The third group of algorithms are based on color scribbles by applying simple strokes in regions of the image. The color of the strokes is then propagated to the remainder of the selected region of the image.

Levin et al. [16] is a pioneer of this method and proposes a simple yet effective user-guided colorization method where the user is required to scribble the desired colors in the interiors of the various regions. These constraints are formulated as a least-squares optimization problem that automatically propagates the scribbled colors to produce a completely colorized image.

Drew et al. [17] present an algorithm for adjusting the color gradients of a colorized image to generate the same maximum-contrast direction as in the original grayscale image, where the contrast in the colorized image may not match the gradient perceived in the original grayscale image.

Quang et al. [3] find that Reproducing Kernel Hilbert Spaces (RKHS) have (both algorithmically and theoretically) recently emerged as a powerful paradigm. The main contribution of this approach is to apply the theory of RKHS and to extend an RKHS-based function to image and video colorization.

Previous work took colorization control out of the hands of the user, requiring intensive computational labor for the user to specify which parts should be colored. Even with such computational costs, the results usually failed on the strong edges. In contrast, we present a fast, simple and efficient algorithm, that can even be used for challenging images, as we will see in the results.

2 Edge-preserving smoothing filter (EPSF)

Smoothing an image while preserving its edges is necessary in a wide variety of applications. Simple smoothing operations such as Gaussian filtering, which do not take into account intensity variations within an image, tend to blur edges.

Several EPSF methods have been proposed to take into account the local structure of images. The best and most popular is the bilateral filtering [5].

2.1 The bilateral filter

The bilateral filter is a nonlinear process that smoothes images while preserving their edges [5]. For an image I, at position s, it is defined by the following:

$$I_s = \frac{1}{W_s} \sum_{p \in \Omega} f(p-s) g(I_p - I_s) I_p \qquad (1)$$

where Ω is the set of image pixels, the subscripts s and p indicate spatial locations of pixels, and W_s is a normalization: $\sum_{p\in\Omega} f(p-s)g(I_p - I_s)$.

The filter output at each pixel is a weighted average of its neighbors. The weight assigned to each neighbor decreases with both the distance in the image plane (the spatial domain f with a standard deviation σ_s) and the distance on the intensity axis (the range domain g with a standard deviation σ_r), using a Gaussian function as a decreasing function.

The main feature of the bilateral filter is its ability to preserve edges while smoothing an input image. Although a naïve implementation of the bilateral filter is costly, due to the joint spatial and range filtering, several fast approximations have been proposed [18,19].

The latest studies have noted, however, that the bilateral filters have limitations, especially on the edges of the objects. We explain these limitations by using the simple test image (Fig. 2a), this image is a piecewise constant function that has been corrupted by zero-mean Gaussian noise with small

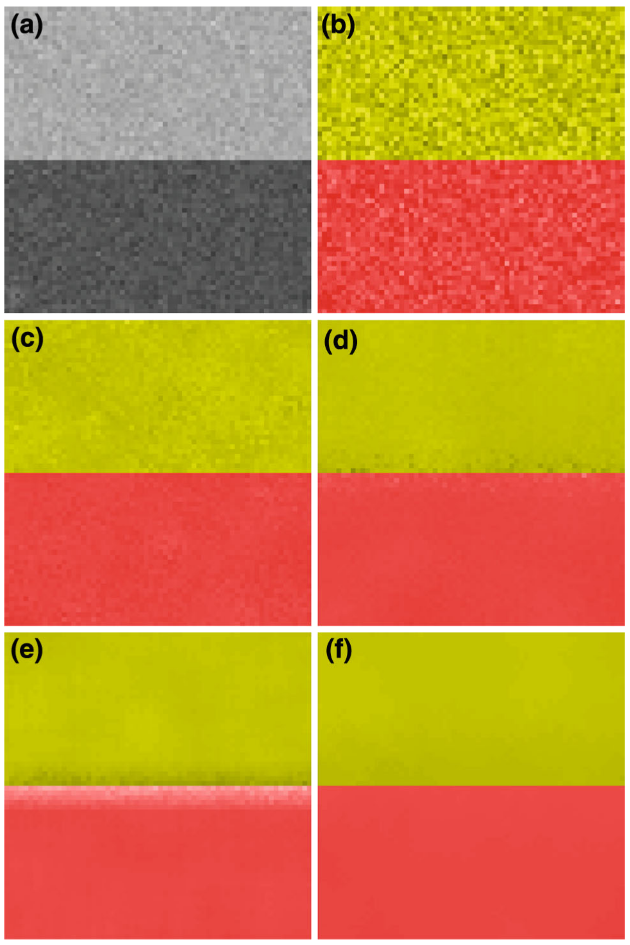

Fig. 2 Filtering test image with the bilateral filter and the Nikolaou filter. **a** Noisy image. **b** Visualization. **c** BF: $\sigma_r = 0.2$, $\sigma_s = 3$. **d** BF: $\sigma_r = 0.2$, $\sigma_s = 13$. **e** RBF: $\sigma_r = 0.6$. **f** NF: $\sigma_r = 0.6$

variance. For clarity, we visualize the image intensities using a color map (shown in Fig. 2b).

First, we attempt to smooth out the noise by applying the bilateral filter with a small σ_s(Fig. 2c). We show that it smoothes the fine-scale noise in the input image while preserving its edge. By increasing only σ_s, the bilateral filter smoothes out large-scale noise (Fig. 2d), but the filtering output has gradient-reversal artifacts. The reason is that when a pixel (often on an edge) has few similar pixels around it, the Gaussian weighted average is unstable. The artifacts increase as σ_s increases. It is shown that the behavior of the bilateral filter becomes a range filter when σ_s is limited. Thus, to get more, smoothing is not only through increased σ_s, but also the range support σ_r must be increased as well. The range weight prevents pixels on one side of a strong edge from influencing pixels on the other side since they have different values.

The primary focus of this paper adopts a fast and robust edge-preserving filter. Because noise is out of the scope of our work, we will be ignoring the spatial smoothing in the future, and we will apply the bilateral filter as a range weights filter.

However, the range bilateral filter blurs colors across edges when σ_r gets larger (Fig. 2e), where the likelihood of two similarly colored pixels in the neighborhood being separated by edge grows quickly. But in many cases, a more restrictive criterion may be better used to describe a given guidance image during the filtering process. For example, in our work, the output chrominance channels should have consistent edges with the given luminance channel.

Nikolaou et al. overcomes these issues by proposing an edge-preserving smoothing operator like the range weights bilateral filter, but having better behavior near the edges. Nikolaou filter does not find the weight average of the pixel differences directly; instead, they extract the Manhattan color distances between the central pixel with its neighbor and they replace the Gaussian function in a range bilateral filter with a robust coefficient function. This function allows us to keep track of how much color propagates to each pixel in the image and normalizes it accordingly. Figure 2f shows the Nikolaou filter results as a comparison with the range weights bilateral filter in Fig. 2e. For both these figures, we set σ_r to 0.6 with 3×3 support. Due to the $((\sigma_r \; or \; k) \geq 1)$ in the Nikolaou formula, we scale the σ_r with Nikolaou filter for testing both figures simultaneously. We show that, for the same running time, the Nikolaou filter is more robust than the range bilateral filter. In our experimental results, we demonstrate these differences using real images.

In the same spirit of the Nikolaou method, Heinrich et al. [20] improve the bilateral filter by dealing with the two Gaussian functions in the bilateral filter separately via a single weighting of the shortest path distance on the image graph and then recombining the two types of distances in one filter.

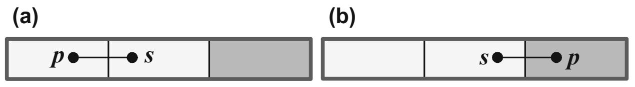

Fig. 3 The pixels value differences in the image regions. **a** The pixel ***s*** and ***p*** in the constant region. **b** The pixel ***s*** and ***p*** in the discontinuous region

2.2 Nikolaou nonlinear filter

Nikolaou et al. [6] introduce a new concept to adopt bilateral filter. They improve bilateral filter by ignoring Gaussian spatial domain while at the same time replacing the Gaussian range domain with a new function. The resulting filter produces more robustness near the edges and performs its task with low computation time.

Assuming that I is a noisy gray-level image. The differences for the value I_s at pixel s with its eight neighbors I_p are normalized in range [0,1]:

$$d_i = \frac{\mid I_p - I_s \mid}{255}, \quad 0 \leq d_i \leq 1, \quad i = 1, \ldots, 8 \tag{2}$$

and then, the function g is computed as follows:

$$g(d(p,s)) = (1 - d_i)^k, \quad \text{where} \quad k \geq 1 \tag{3}$$

where the factor k scales exponentially the intensity differences, it controls the amount of blurring performed on the image. This concludes to the following nonlinear smoothing filter:

$$I_s = \frac{1}{W_s} \sum_{p \in \Omega} g(d(p,s)) I_p \tag{4}$$

where Ω is the set of image pixels, the subscripts s and p indicate spatial locations of pixels, and W_s is a normalization: $\sum_{g(d(p,s))}$.

To give a statistical interpretation for the Nikolaou filter, we set the function g as a "weight" and from there enlarge a part of Fig. 2 (shown in Fig. 3a). The differences of intensity value between pixel s and its neighbor pixel p in the same(originally constant) region will be small, and hence, g will be heavily weighted. This is equivalent to choosing an s value to be the mean of the neighboring intensity values. In the next section, our algorithm will use this property for expanding the color in all directions within a specific region. This is not true for pixels across edges (Fig. 3b). Due to the image features (discontinuity of intensity) and the estimated intensity value at pixel s on one side of an intensity discontinuity, measurements from the value of pixel p at the other side should be "rejected". In the context of the Nikolaou filter, the function g will be lightly weighted and has only a small effect on the final result. Hence, each iteration in Eq. 4 will produce only a small change in the image.

3 Colorization

We present our method in the YUV color space [21]. The Y component represents the luminance of the pixels in a grayscale image, while the U and V components correspond to the chrominance (color information) as provided by the user via color hints. Let Ω represent the image domain. Our algorithm assumes that the complete grayscale image is given in the entire Ω. Let the small patches where the color is given be the domain Ω_c such that $|\Omega_c| \ll |\Omega|$. The object from the knowledge of Y in Ω and (U, V) in Ω_c is to colorize the whole domain Ω.

In our formula, we calculate the function g for every value in the Y component by finding the differences between each central value s with eight neighboring values p in a 3×3 window:

$$d_i = \mid Y_p - Y_s \mid /255, \quad 0 \leq d_i \leq 1, \quad i = 1, \ldots, 8 \tag{5}$$

and then the function g is computed as:

$$g(d(Y_{p,s})) = (1 - d_i)^k \tag{6}$$

where k controls the amount of colors that will diffuse across edges. A fixed value $k = 30$ is used for all our experiments. In this section, we modified the Nikolaou formulation, and we introduce a fast and stable nonlinear colorizing filter(Eq. 7) to solve the problem of colorizing grayscale images:

$$(U, V)_s^{t+1} = \frac{1}{\sum_{g(d(Y_{p,s}^t))}} \sum_{p \in \Omega} g(d(Y_{p,s}^t))(U_p^t, V_p^t) \tag{7}$$

where (U_p, V_p) represents the color hints and t denotes discrete time steps(iterations). In our experimental results, we found that between 700 and 1000 iterations t reached the stability of Eq. 7 and produced good results.

Algorithm 1 shows the steps of our algorithm. The idea is that the colorizing process obtained by our algorithm is "conditional": When (U_p, V_p) has color information and the function g within a piecewise constant image region, the expanding of color information (U, V) will increase with iterations.

On the other hand, when the function g is applied to boundary pixels (edge pixels), it will be lightly weighted, and hence, the colors of these neighboring pixels have a small effect on the final color received by the central pixel. For this reason, our algorithm will consider these pixels as edge.

Algorithm 1: Colorizing grayscale images

```
   Input: Grayscale image with hints image
   Output: Colorized image
   // Initialization
 1 $k \leftarrow 30$;
 2 $iter \leftarrow 1000$ ; // The number of iterations,
   usually between 700 and 1000. We assume
   $iter = 1000$
 3 foreach $pixel\ s \in hints\ image$ do
 4     $(YUV)_s \leftarrow Convert((RGB)_s)$;
 5     if $(U, V)_s$ has hint then
 6         $flag \leftarrow True$
 7     else
 8         $flag \leftarrow False$;
   // Compute the range weights function $g_i$
 9 foreach $pixel\ s \in grayscale\ image$ do
10     $d_i \leftarrow \frac{|Y_p - Y_s|}{255}$ ; // $0 \leq d_i \leq 1$, $i = 1..8$
11     $g_i(d(Y_{p,s})) \leftarrow (1 - d_i)^k$;
   // Colorization
12 for $t \leftarrow 0$ to $iter - 1$ do
13     if $flag = False$ then
14         $(U, V)_s^{t+1} \leftarrow \frac{1}{\sum_{g(d(Y_{p,s}^t))}} \sum_{p \in \Omega} g(d(Y_{p,s}^t))(U_p^t, V_p^t)$;
15     else
16         $(U, V)_s^{t+1} \leftarrow (U, V)_s^t$;
17     $(U, V)_s^t \leftarrow (U, V)_s^{t+1}$;
18 $(RGB)_s \leftarrow Convert((YUV)_s)$;
```

4 Results and limitations

The algorithm is implemented and tested using an *Intel®* *Core^{TM} 2Duo(E7500@2.93 GHz)CPU* and *3GB RAM PC*. We currently do not use a Graphical Processing Units (GPU) for the implementation of our algorithm. The choice whether to use the CPU or the GPU is driven not only by efficiency issues; in fact, CPUs are still far more prevalent than GPUs (e.g., mobile devices and many laptops have CPUs but no GPUs). The large diversity between GPUs also makes it difficult to deploy GPU-based software to a wide audience [22]. In addition, a large number of algorithms fail to achieve satisfactory performance gain, since the inherent nature of the algorithms and the GPU platform is uncooperative [23]. The proposed algorithm is tested on several challenging images from the Berkeley segmentation [24], previous work and full-HD images. The algorithm almost always stopped between 700 and 1000 iterations.

We start with an evaluation of algorithm performance on a human face and show its superiority over state-of-the-art methods in both accuracy and efficiency. The most difficult problems in facial colorization are smoothly colorizing the skin and keeping the seamless connections among hair, skin, lip, eye, eyebrow and background. Figure 4a (321 $\times$ 481

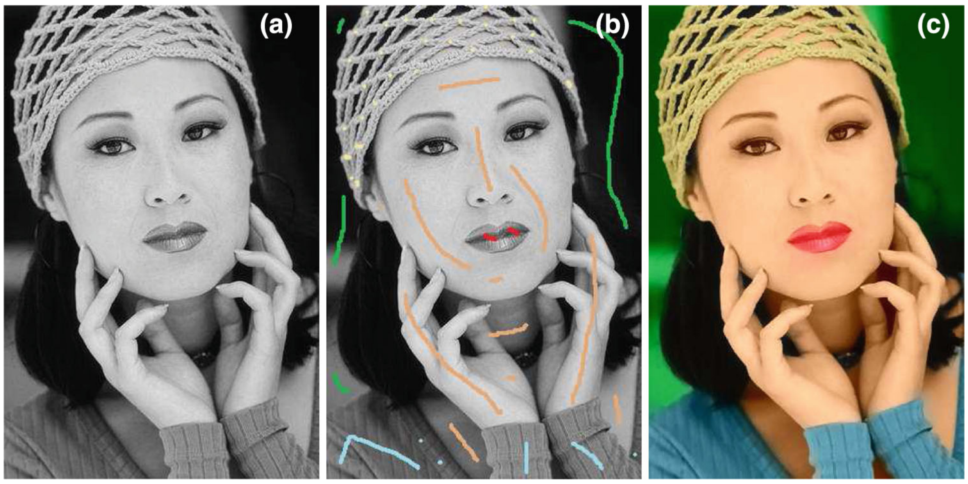

Fig. 4 Girl image colorization example. **a** The *grayscale* image. **b** The *grayscale* image marked with some color scribbles by the user. **c** The colorized image. The image size/run time 321 × 481/less than 0.11 sec and t = 700

pixels) shows the grayscale image of the girl's face and in the Fig. 4b, the image marked with some color scribbles by the user. In Fig. 4c, we obtained the result with a run time less than 0.11 sec and t=700. The colorized result has more visual appeal than the original grayscale image while keeping the delicate details of the object boundaries.

To evaluate the capabilities of our technique with others, we compare our results with three representative colorization algorithms. We have chosen the Liven et al. [16] algorithm, the color transfer technique by Welsh et al. [8], Pouli et al. [15], and finally, we compare our technique with the range weights bilateral filter.

Levin et al. [16] propose an interactive colorization technique based on the premise that nearby pixels in space that have similar gray levels should also have similar colors. However, because no boundary or region information is considered in their colorization process, color sometimes diffuses from one region to others (Fig. 5c). In contrast, because we take edges into accounts, our proposed algorithm will keep the "edges" efficiently, as shown in Fig. 5d.

Figure 6 shows the comparison with the previous work. The input images[1](source, target) in the Fig. 6a,b are used with the methods of Welsh et al. [8] and Pouli et al. [15] for the purpose of transferring colors from the target colored image to the source grayscale image.

Figure 6d shows the result of the Welsh et al. algorithm. In general, the problem with color transfer approaches is that if the contents of the target image are different than the source, the results can look unnatural. For instance, the colors of the target image are successfully transferred to the source in Fig. 6d, but the background of the target image contains some unwanted red patches. In addition, color transfer may require additional controls to specify which parts of the source image should be colored or which parts of the target

[1] The images are adopted from Pouli et al. [15] and reprinted with permission from Elsevier

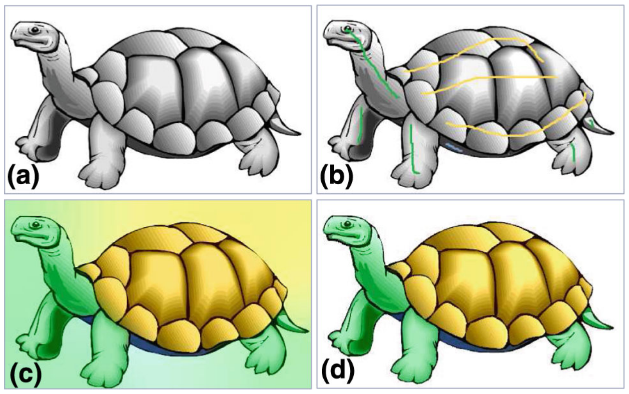

Fig. 5 Comparison with Levin et al. algorithm. Levin et al.' algorithm fails on strong edges of images. **a** The *grayscale* image. **b** The image marked with some color scribbles. **c** Levin et al. [16]. **d** Proposed algorithm

should be considered. Pouli et al. [15] used corresponding mattes to produce the result shown in Fig. 6e. They used a mask that only includes the flower; the yellows of the target are only transferred to the flower in the source. Although they obtained satisfactory results, image matting is known to be a hard problem in computer vision. The run time of the two above algorithms for the 680 × 455 pixel is 241.836 sec. We note that the above methods are very time-consuming.

We compare our filter with the range weights bilateral filter (bilateral filter without Gaussian spatial domain) by colorizing the same input image, the marked image shown in the Fig. 6c. We colorize the input image by using the range weights bilateral filter (Fig. 6f) and by using our algorithm (Fig. 6g) with $\sigma_r = 0.3$. In (Fig. 6g), we scale σ_r to correspond to parameter k in the Nikolaou filter. Comparison of the visual quality of Fig. 6f,g indicates that the bilateral filter is unable to preserve edges efficiently and some of them become blurred, which may produce halo artifacts, while our algorithm effectively enhances the quality of the resulting image (especially on the edge) and improves the performance of the bilateral filter. The run time of our result is 0.34 sec.

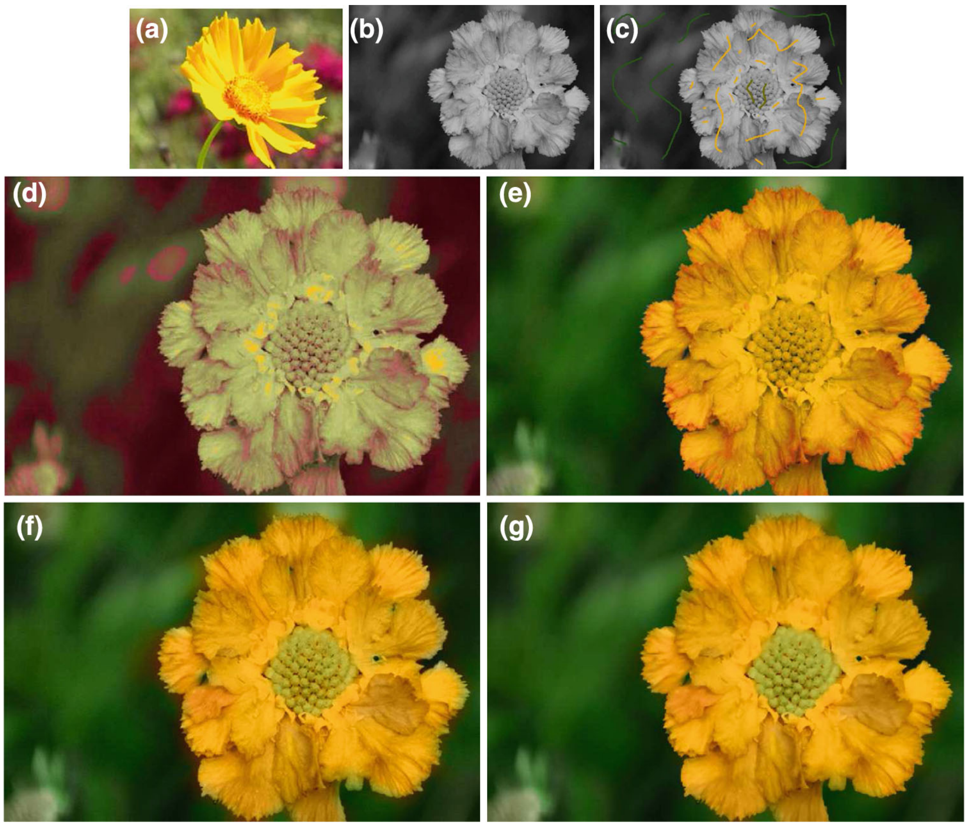

Fig. 6 Comparison with the previous work. **a–c** The input images. **d** Welsh et al. [8]. **e** Pouli et al. [15]. **f** Range bilateral filter. **g** Proposed algorithm

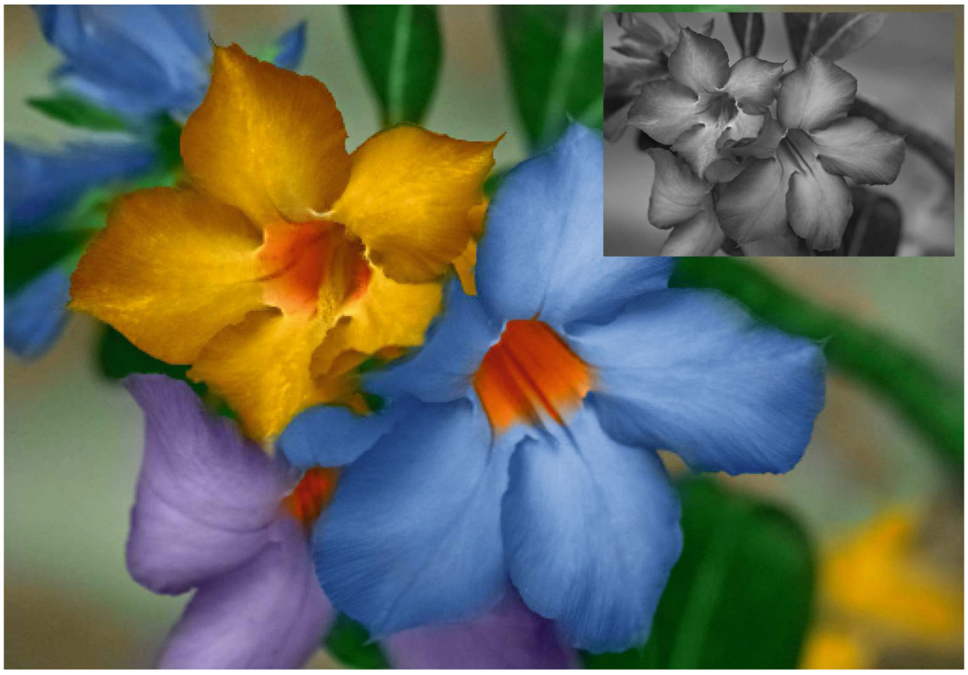

Fig. 7 Full-HD image colorization example. The input monochromatic image on the *top right*. Then, the colorized result shown with downsampling. The image size/run time 1, 024 $\times$ 709 pixels/ 0.61 sec

Note that we reduce the process duration as compared with the run times of the Welsh et al. and Pouli et al. methods. Stability of the colorizing image is obtained by a proper setting of the σ_r as compared with the bilateral filter.

In Fig. 7, we perform a similar experiment with a full-HD image.[2] A direct implementation of our filter may be take

[2] The image is publically available at:http://www.f11digital.com/

several minutes; therefore, we downsample the input image using Nearest-neighbor algorithm; the simplest method is just to replace the pixel value with the nearest neighboring pixel. One possible reason to use Nearest-neighbor algorithm is that, unlike most other algorithms such as bicubic spline, bicubic B-spline and Lanczos, it is most suitable for reprojecting an image (without a change in pixel size) which when preserving the original pixel values (color hints) for later operations is important. Other above algorithms are not well suited for our operations. Figure 7(top right) shows the input image; a monochromatic image, a colorized result shown on the same figure(with downsampling). The image size/run time is 1024×709 pixels/1.13 sec (without downsampling) and 0.61 sec (with downsampling by a factor 0.5). The result shows more robustness and higher speed when compared to previous available approaches.

Limitations Although the proposed method produces convincing results on a variety of images, there is a limiting factor that should be taken into account, as it can lead to unexpected behavior; if the placed scribble is located on the wrong side of the edge or with high gradients between it, the resulted image will suffer from color artifacts visible, especially near edges, since the color will diffuse from one region to others. Sometimes, this mistake is intentional by the user. For complex images, as shown earlier in Fig. 4, enough tightly positioned scribbles along the object's boundaries must be given for producing high-quality color images, so that the user's effort is minimized.

Of course, the proposed method is realized by an interactive method, the robustness and speed of the algorithm allows the user to fix such mistakes and to add or move color seeds as needed.

5 Conclusions

In this paper, we present a new algorithm for interactive colorization by introducing the Nikolaou nonlinear filter. Nonlinear filters that preserve edges are a common technique for removing noise from images. There are many filters for removing noise; some works use these filters, such as the bilateral filter, for removing unnecessary details from images to communicate intended information and use abstraction for effective visual communication. In our algorithm, instead of removing noise or unnecessary details from the images, we are using the Nikolaou filter to automatically expand the color hints from the user's paint brush and align them with the object boundary. Experimental results show that our algorithm works effectively on several challenging images and succeeds in areas where previous methods have failed.

Acknowledgments The authors would like to thank the anonymous reviewers for their comments. The work is partially supported by the National Basic Research Program of China (973 Program) (No.2012CB316201), the National Natural Science Foundation of China (Nos. 60973018, 60973020), the Doctoral Fund of Ministry of Education of China (No. 20110042110028), and the Fundamental Research Funds for the Central Universities (No. N110804002).

References

1. Yatziv, L., Sapiro, G.: Fast image and video colorization using chrominance blending. In: IEEE Trans. Image Process. **15**(5), 1120–1129 (2006)
2. Vieira, L.F.M., do Nascimento, E.R., Fernandes Jr., F.A., Carceroni, R.L., Vilela, R.D., Araújo, A. de A.: Fully automatic coloring of grayscale images. Image Vis Comput. **25**(1), 50–60 (2007)
3. Quang, M.H., Kang, S.H., Le, T.M.: Image and video colorization using vector-valued reproducing Kernel Hilbert spaces. J Math. Imaging Vis. **37**(1), 49–65 (2010)
4. Fornasier, M., Ramlau, R., Teschke, G.: The application of joint sparsity and total variation minimization algorithms in a real-life art restoration problem. Adv. Comput. Math. **31**(1–3), 157–184 (2009)
5. Tomasi, C., Manduchi, R.: Bilateral filtering for gray and color images. In: Proceedings 6th International Conference Computer Vision. Bombay, India, Jan 4–7, pp 839–846 (1998)
6. Nikolaou, N., Papamarkos, N.: Color reduction for complex document images. Int. J. Imaging Syst. Technol. **19**(1), 14–26 (2009)
7. Markle, W., Hunt, B.: Coloring a Black and White Signal Using Motion Detection. US Colorization Inc. Canadian patent no.1291260, Dec. (1987)
8. Welsh, T., Ashikhmin, M., Mueller, K.: Transferring color to grayscale images. ACM Trans. Graphics **21**(3), 277–280 (2002)
9. Eisemann, E., Durand, F.: Flash photography enhancement via intrinsic relighting. ACM Trans. Graphics **23**(3), 673–678 (2004)
10. Wang, J., Cohen, M.F.: Image and video matting: a survey. Trends Comput. Graph. Vis. **3**(2), 97–175 (2007)
11. Levin, A., Lischinski, D., Weiss, Y.: A closed-form solution to natural image matting. In: IEEE Trans. Pattern Anal. Mach. Intell. **30**(2), 228–242 (2008)
12. He, K., Sun, J., Tang, X.: Fast matting using large kernel matting laplacian matrices. In: Proceedings 23rd IEEE Conference Computer Vision and Pattern Recognition. San Francisco, USA, Jun. 13–18, pp. 2165–2172 (2010)
13. Charpiat, G., Hofmann, M., Scholkopf, B.: Automatic image colorization via multimodal predictions. In: Proceedings 10th European Conference Computer Vision. Marseille, France, Oct. 12–18, pp. 126–139 (2008)
14. Kekre, H.B., Thepade, S.D., Bhandari, N.: Colorization of grayscale images using Kekre's biorthogonal color spaces and Kekre's fast codebook generation. Adv. Multimed. Int. J. **1**(3), 49–58 (2011)
15. Pouli, T., Reinhard, E.: Progressive color transfer for images of arbitrary dynamic range. Comput. Graphics **35**(1), 67–80 (2011)
16. Levin, A., Lischinski, D., Weiss, Y.: Colorization using optimization. ACM Trans. Graphics **23**(3), 689–694 (2004)
17. Drew, M.S., Finlayson, G.D.: Improvement of colorization realism via the structure tensor. Int. J. Image Graphics **11**(4), 589–609 (2011)
18. Chen, J., Paris, S., Durand, F.: Real-time edge-aware image processing with the bilateral grid. ACM Trans. Graphics **26**(3), pp 103:1–10 (2007)
19. Paris, S., Durand, F.: A fast approximation of the bilateral filter using a signal processing approach. Int. J. Comput. Vis. **81**(1), 24–52 (2009)

20. Heinrich, S.B., Snyder, W.E.: Improved Edge Awareness in Discontinuity Preserving Smoothing. arXiv, Vol. arXiv:1103.5808, (2011) Available at: http://arxiv.org/abs/1103.5808
21. Jack, K.: Video Demystified: A Handbook for the Digital Engineer(3rd) Ch.3, pp. 15–17. LLH Technology Publishing, Eagle Rock (2001)
22. Criminisi, A., Sharp, T., Rother, C., Pérez, P.: Geodesic image and video editing. ACM Trans. Graphics **29**(5), 1–15 (2010)
23. Park, K., Singhal, N., Lee, M.H., Cho, S., Kim, C.W.: Design and performance evaluation of image processing algorithms on GPUs. In: IEEE Trans. Parallel Distrib. Syst. **22**(1), 91–104 (2011)
24. Martin, D.R., Fowlkes, C., Tal, D., Malik, J.: A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In: Proceedings 8th International Conference Computer Vision vol. 2. Vancouver, British Columbia, Canada, July 7–14, pp. 416–425 (2001)