Image Anal Stereol 2021;40:17-27 doi: 10.5566/ias.2348
Original Research Paper

SMOOTHING AND CLUSTERING GUIDED IMAGE DECOLORIZATION

FANG LI✉ AND YUANMING ZHU

School of Mathematical Sciences and Shanghai Key Laboratory of PMMP, East China Normal University, Shanghai, China
e-mail: fli@math.ecnu.edu.cn, ymzhu385@126.com

(Received February 18, 2020; revised December 8, 2020; accepted February 2, 2021)

ABSTRACT

In this paper, we propose a new image decolorization method based on image clustering and weight optimization. First, we smooth the color image and cluster it into several classes and get the class centers. Each center can represent a distinctive color in the image. Then the class centers are sorted according to their brightness measured by Euclidean norm. By assuming that the decolorized grayscale image is a linear combination of the three channels of the color image, we propose an optimization problem by forcing the sorted class centers to correspond to specified grayscale values satisfying uniform distribution. Numerically, the problem is solved by quadratic programming. Experiments on two popular data sets demonstrate that the proposed method is competitive with the state-of-the-art decolorization method.

Keywords: Clustering, contrast, decolorization, optimization, smoothing.

INTRODUCTION

Image decolorization, also known as color-to-gray, aims to convert a color image into a grayscale one. It is widely used in single-channel image and video processing, digital printing, and photography. Decolorization maps 3D color space into 1D space, such that the information loss is inevitable. Hence the main focus of decoloration is preserving the salient features such as edges and contrast.

The existing decolorization methods can be roughly categorized into global and local methods. Local methods treat the pixels with the same color differently in order to enhance local chrominance edges. Bala and Eschbach (2004) enhanced the color edges by adding high-frequency components of chromaticity to the lightness channel. Neumann *et al.* (2007) introduced an efficient gradient-based color to gray transformation algorithm based on Coloroid color space. Smith *et al.* (2008) proposed a two-step decolorization method that combines a global mapping based on perceived lightness with local chromatic contrast enhancement. Gooch *et al.* (2005) proposed to minimize the differences between the chrominance and luminance values of pixel pairs. Jin *et al.* (2014) proposed a variational approach by maximizing variance. These local methods are good at enhancing local contrast; however, the disadvantage is that they may occasionally distort the appearance of constant color regions Kim *et al.* (2009).

Global methods apply a constant mapping function on all the image pixels such that pixels of the same color are mapped into the same grayscale value. Rasche *et al.* (2005) used a linear color mapping function to obtain the optimal conversion by imposing constraints on different color pixel pairs. Grundland and Dodgson (2007) proposed a parametric piecewise linear mapping algorithm for image decolorization by adjusting the grayscale value with the chrominance. Kim *et al.* (2009) proposed a non-linear global mapping function for decolorization and estimated the parameters by minimizing the color differences. Lu *et al.* (2012a) employed a bimodal energy function to raise a more flexible contrast preserving constraint. Then Lu *et al.* (2012b) proposed a real-time decolorization algorithm by imposing constraints on weights based on the model in Lu *et al.* (2012a). The limitation of global methods is that they cannot fully capture the details in a globally unified manner, which may lead to local contrast loss.

Many methods use both global and local information. Lu *et al.* (2014) improved their model in Lu *et al.* (2012a) by introducing local and non-local constraints. Du *et al.* (2015) proposed a saliency-guided decolorization method using region-based optimization, in which the saliency combines both the local and global features of the images. Wang *et al.* (2018) proposed a global mapping to achieve a fast computation on color order, and then a local decolorization algorithm is designed on the basis of the global linear mapping so that both color and spatial information are preserved. Zhao *et al.* (2018) proposed a new multimodal contrast-preserving measure with a multimodal Gaussian distribution to relax the constraint of color contrast, which preserves both the local color contrast and the non-local color contrast. Ji *et al.* (2016) introduced a new band-pass filter for color-to-gray conversion. Liu *et al.* (2015; 2019; 2017a;b) proposed several effective methods for image decolorization. Liu *et al.* (2015) proposed a gradient correlation similarity measure-based decolorization model (GcsDecolor). Liu *et al.* (2017a) proposed a semiparametric decolorization method (SPDecolor) by decomposing the first-order and the two-order color space. Liu *et al.* (2019) presented a weighted projection maximum function to model the decolorization procedure. Liu *et al.* (2017b) introduced the Log-Euclidean metric for decolorization.

Recently, deep learning was introduced to image decolorization. Hou *et al.* (2017) proposed a convolutional neural network for image decolorization with a perceptual loss. Zhang and Liu (2018) proposed a contrast preserving image decolorization method based on convolutional neural network which combining global features and local semantic features. Liu and Zhang (2019) designed a new convolutional neural network framework combining both local features and exposure features of the color image. Liu and Leung (2018) proposed a variable augmented neural network for decolorization.

Clustering of the color image is included in some methods Lau *et al.* (2012); Du *et al.* (2015). Lau *et al.* (2012) proposed to cluster the image and then construct a graph based on each segment for decolorization. The method is semi-local because it operates on clusters, modifying local contrasts between clusters. Du *et al.* (2015) proposed to use the clustered superpixels to define region contrast saliency in their decolorization energy. It contributes to the region contrast preserving in the decolorization process. Experimentally, these methods work well on some test images. However, the overall performance on large datasets is still limited.

In this paper, we propose a novel smoothing and clustering guided decolorization method to reduce contrast loss in the global method. As is known, we cannot assign different gray values for different colors if the number of different colors is larger than the number of all the gray values. However, there are usually not many distinctive colors in a specific image since many pixels have similar colors. This observation motivates us to assign distinctive gray values for only distinctive colors. Our method has three steps. Firstly, the color image is smoothed by L_0 smoothing method such that the colors are largely reduced. Then the smoothed image is segmented into several regions by using a revised fast fuzzy C-means method. The class centers represent distinctive colors, which are sorted according to their Euclidean norm. Then we form an optimization problem by requiring that the class centers are mapped into grayscale values satisfying uniform distribution. The parametric linear mapping is adopted for color-to-gray conversion. Experiments and comparisons are conducted on two popular datasets to verify the effectiveness of our method.

OUR METHOD

We describe in this section our decolorization method. In Fig. 1, we show the flowchart of our method by displaying all the main intermediate steps. The proposed method includes three steps: smoothing, clustering, and decolorization. In the following, we show the details of each step.

SMOOTHING

Smoothing is a preprocess for many image problems. In our method, the aim of using smoothing is to reduce the colors of the given color image. As we know, RGB color space is 3D and contains 256^3 colors generally. Natural images usually contain tens of thousands of colors. Through the smoothing step, the number of colors can be largely reduced, which is useful for accelerating the next clustering step.

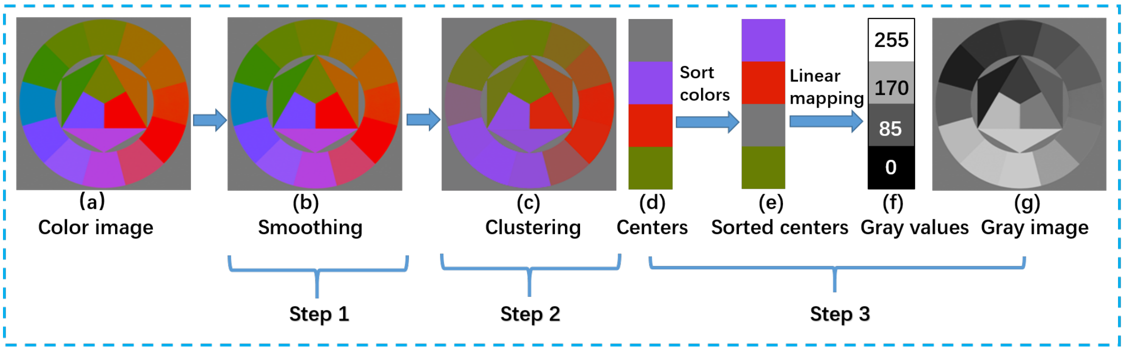


Fig. 1: The flowchart of our method. Fig. 1a is from Cadík's dataset Cadík (2008).

There exist many smoothing methods in the literature; some representative methods are Xu *et al.* (2011); Min *et al.* (2014); Ham *et al.* (2017). Xu *et al.* (2011) proposed an L_0 gradient minimization method for image smoothing, which can preserve sharp edges globally (L_0 smoothing for short)). Min *et al.* (2014) proposed a fast global image smoothing method based on weighted least squares (WLS for short). Ham *et al.* (2017) proposed a novel SD (for static/dynamic) filter for image smoothing, which jointly leverages structural information from guidance and input images (SDF for short). We choose L_0 smoothing method Xu *et al.* (2011) as a baseline in this step. Noting that choosing other smoothing techniques also works. The influence of different smoothing methods in our method will be quantitatively evaluated in the experimental section.

The L_0 smoothing method can be described as follows. Assume I_0 is the given image. L_0 smoothing aims to solve the following problem:

$$\min_{I} \lambda \|\nabla I(x)\|_0 + \|I(x) - I_0(x)\|_2^2, \tag{1}$$

where $x \in \Omega$ and Ω is the 2D image region. The first term is the L_0 regularization term, which requires that the number of pixels with non-zero gradient is as small as possible. The second term is the fidelity term, which requires that the smoothing result is close to the given image. L_0 smoothing leads to cartoon-like images with sharp edges. In the decolorization process, these sharp edges are the key features that should be preserved.

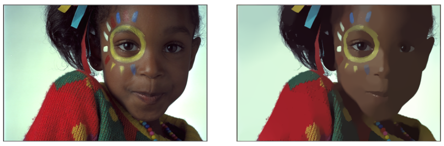


Fig. 2: Smoothing of a natural color image, $\lambda = 0.01$. (a) Original color image with size 260×390 (101400 pixels, 40582 different colors); (b) the result of L_0 smoothing (15577 different colors).

In Fig. 2, we show the L_0 smoothing result of a natural image in Cadík's dataset Cadík (2008). It can be seen that the main content and sharp edges are preserved in Fig. 2b, while the colors are reduced from 40582 to 15577. That is, 62% of the colors are reduced. For cartoon-like images, the smoothing effect is not so obvious. See Fig. 1, for example. After smoothing, the colors reduce from 2783 to 2114. However, cartoon images are much easier to segment than natural images since the number of colors in cartoon images is usually small.

CLUSTERING

Image clustering aims to partition an image into several disjoint regions such that pixels in the same region share some uniform characteristics such as intensity, color, and texture. Fuzzy c-means (FCM) clustering method Bezdek *et al.* (1984) is a widely used method for image clustering. The standard FCM model for partitioning image I into N class is given by

$$\min \sum_{i=1}^{N} \int_{\Omega} (I(x) - c_i)^2 u_i^2(x) dx \tag{2}$$

where Ω is the image region, $\{c_i\}_{i=1}^N$ are the class centers and $\{u_i\}_{i=1}^N$ are membership functions satisfying some constraints. The computational complexity of the FCM algorithm increases as the image size increases.

Szilagyi *et al.* (2003) proposed a faster enhanced FCM algorithm for magnetic resonance (MR) image segmentation. The basic idea is constructing a new image based on the given MR image and then performing the FCM in the range domain of the new image. Since they consider gray images, the range domain is 1D. In this paper, we generalize their idea to color images. The range domain of a color image is 3D. Assume that the range domain contains L different colors. The variable in the range domain is denoted as $\xi_r \in [0,255]^3, r = 1,...,L$ where r is the index. Then we minimize the following energy function:

$$\min \sum_{i=1}^{N} \sum_{j=1}^{3} \sum_{r=1}^{L} v_r (\xi_{rj} - c_{ij})^2 u_{ir}^2 \tag{3}$$

where v_r is the number of pixels with the same color ξ_r, which can be seen as the weight of ξ_r, j denotes color channel, c_i denotes the i-th class center vector. The numerical algorithm is similar to the FCM algorithm, so we omit the details. The revised method is much faster than the standard FCM for color image clustering, so we call this method as fast FCM (FFCM).

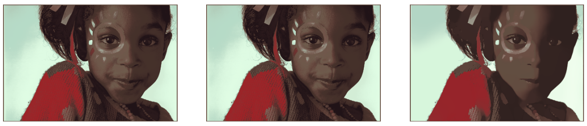


Fig. 3: Clustering of color images in Fig. 2, class number is $N = 6$. (a) Result of the standard FCM on the original image in Fig. 2a, computational time=3.01s; (b) result of FFCM on the original image in Fig. 2a, computational time=0.86s; (c) result of FFCM on the smoothed image in Fig. 2b, computational time=0.26s.

In Fig. 3, we compare the clustering results of FCM and FFCM. Note that for FCM, we show $\sum_i^N c_i u_i(x)$ as the final clustering results. For FFCM, we let the membership value of pixel x in class i as u_{ir} where $I(x) = \xi_r$. Then the membership functions $\{\tilde{u}_i(x)\}_{i=1}^N$ can be defined for each pixel x such that the final clustering is given by $\sum_i^N c_i \tilde{u}_i(x)$. Visually, the result of FCM and FFCM are similar; see Fig. 3a and Fig. 3b. The result of FFCM on the smoothed image in Fig. 3c seems good. Meanwhile, FFCM saves 72% computational time than FCM. When applying FFCM on the smoothed image, we can save another 70% computational time. On the whole, our FFCM together with L_0 smoothing is about 11 to 12 times faster than FCM without smoothing.

In order to evaluate the performance of FCM and FFCM on image segmentation, we use the standard test color images set from the Berkeley Segmentation Dataset BSDS500 `https://www2.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/.`, which contains 100 color images in the test folder. The segmentation number is fixed as 10. The PSNR of FCM and FFCM segmentation results are 29.25dB and 29.29dB, respectively. The total computational time of FCM and FFCM are 1575s and 195s, respectively.

Actually, as a preprocessing step for our proposed decolorization method, FCM and FFCM yield similar final decolorization results; see Table 1 in the experimental section. However, with FFCM, the algorithm is much faster than using FCM.

CLUSTERING-GUIDED DECOLORIZATION

Our motivation is that we expect the distinctive colors can be decolorized into distinctive gray values. For example, in Fig. 1, the image is segmented into four classes. The color values of the four cluster centers are displayed in Fig. 1d. These four colors can be seen as distinctive colors in the given color image. Our aim is to assign four distinctive gray values to these four colors. Intuitively, the four distinctive gray values should be equally distributed in [0,255]. Actually, it has been proved that the uniformly distributed gray image has the best contrast Bertalmio *et al.* (2007). Hence we choose the four gray values as 255, 170, 85, 0, see Fig. 1f. The remaining problem is how to correspond the four colors to the four gray values. As an output of the FFCM clustering method, the class centers have no order in Fig. 1d. So we sort the colors according to the brightness of the colors. We choose to measure the brightness of the colors by calculating the Euclidean norm of the color vector. Then we get the sorted colors in Fig. 1e, where brightness decreases from top to bottom. Then we have constructed the correspondence of the four colors in Fig. 1d with the four gray values in Fig. 1f.

Let us go back to our decolorization problem. Following the existing global decolorization methods, we assume that the decolorized gray image g can be represented by a linear combination of the three channels of the color image, i.e.,

$$g(x) = w_1 R(x) + w_2 G(x) + w_3 B(x) \qquad (4)$$

where R, G, B are the red, green and blue channels of the given color image I, and $w_i, i = 1,2,3$ are three scalar weights. The decolorization problem is to find the optimal weights. Assume the class number is $N > 1$. After clustering, we get N sorted colors which can be formulated as an $N \times 3$ matrix $\mathbf{A}$. Each row of $\mathbf{A}$ denotes a color vector. As stated above, we expect that the sorted colors correspond to uniformly distributed gray values in [0,255] one by one. The gray value vector can be written as

$$\mathbf{b} = [255, (N-2)a, (N-3)a, \cdots, 2a, a, 0]^T, \qquad (5)$$

where a is the maximum integer less than $255/N$. For example, in Fig. 1e, we have $N = 4$, $a = 85$ and then $\mathbf{b} = [255, 170, 85, 0]^T$.

To force the correspondence of the colors in matrix $\mathbf{A}$ with the gray values in vector $\mathbf{b}$, based on equation (4), we propose to optimize the weight $\mathbf{w} = [w_1, w_2, w_3]^T$ by solving the following least square problem with constraints:

$$\begin{aligned} &\min_{\mathbf{w}} \|\mathbf{A}\mathbf{w} - \mathbf{b}\|_2^2 \\ &s.t. \;\; 0 \le w_i \le 1, i = 1,2,3, \\ &\qquad w_1 + w_2 + w_3 = 1. \end{aligned} \qquad (6)$$

This minimization problem can be easily solved by quadratic programming with the MATLAB routine ”quadprog”. Then the final decolorization result g is given by formula (4) with the optimal weights determined by (6).

As an example, in Fig. 1, we set the class number as $N = 4$, then we have

$$\mathbf{A} = \begin{bmatrix} 144.42 & 73.99 & 237.98 \\ 224.94 & 31.44 & 4.94 \\ 118.87 & 118.89 & 120.37 \\ 104.37 & 125.94 & 2.65 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} 255 \\ 170 \\ 85 \\ 0 \end{bmatrix}.$$

By calling “quadprog” in MATLAB with inputs $\mathbf{A}, \mathbf{b}$ and constraints, we get the solution

$$\mathbf{w} = \left[0.5111, 0.0000, 0.4889\right]^T.$$

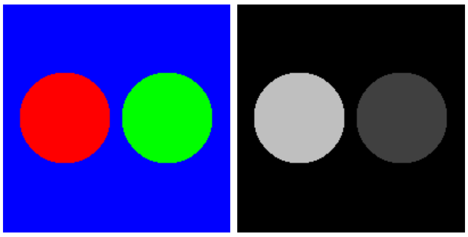
Finally, the output gray image is given by the linear combination of the three channels of the color image with weights 0.5111, 0 and 0.4889, which is shown in

Fig. 1g. It is obvious that distinctive colors in Fig. 1a is also distinctive in the gray image Fig. 1g.

Remark that in our method the colors of the class centers are ordered according to their brightness measured by Euclidean distance. If their brightness are the same, no ordering is possible. In this case, we omit the color ordering step. The minimization problem in (6) can also be solved by quadratic programming. For example, we construct an RGB image with only red (255,0,0), green (0,255,0), and blue (0,0,255) colors in Fig. 4a. Then we have

$$\mathbf{A}=\begin{bmatrix}255 & 0 & 0\\ 0 & 255 & 0\\ 0 & 0 & 255\end{bmatrix},\ \mathbf{b}=\begin{bmatrix}255\\ 127.5\\ 0\end{bmatrix}.$$

The solution of (6) is given by $\mathbf{w}=[0.75,0.25,0]^T$. The decolorized image Fig. 4b shows that our method can distinguish the three colors in the gray image.



(a) color image (b) our result (N=3)

Fig. 4: Test on a synthetic image with uniform brightness. (a) The color image with pure R, G, and B colors; our final decolorization result ($N = 3$).

EXPERIMENTAL RESULTS

In this section, we conduct extensive experiments to evaluate our method and compare it with some representative local and global decolorization approaches using the benchmarking Cadík's dataset Cadík (2008) and COLOR250 dataset Lu *et al.* (2014). Cadík's dataset contains 24 color images, and most of them are synthetic images. COLOR250 contains 250 color images, including natural images, digital charts, logos, and illustrations. We compare our method with eight existing methods CIE-Y, Gooch05 Gooch *et al.* (2005), Smith08 Smith *et al.* (2008), Lu12 Lu *et al.* (2012a), Gcs15 Liu *et al.* (2015), Du15 Du *et al.* (2015), SP17 Liu *et al.* (2017a), and Wpm17 Liu *et al.* (2019). Note that there are two algorithms GcsDecolor1 and GcsDecolor2 in Liu *et al.* (2015), we choose the better one GcsDecolor2. Similarly, there are two algorithms WpmDecolor1 and WpmDecolor2 in Liu *et al.* (2019), we choose the better one WpmDecolor2.

For methods CIE-Y, Gooch05 and Smith08, their results on Cadík's dataset are provided in Cadík (2008), while their results on COLOR250 are provided in Lu *et al.* (2014). The results of Lu12, Gcs17, Du15, SP17 and Wpm17 are either provided by the authors or generated by their source codes with default parameters. Our method is performed under Windows 10 and MATLAB R2018a with Intel Core i7-8500 CPU@1.80GHz @1.99GHZ and 32GB memory.

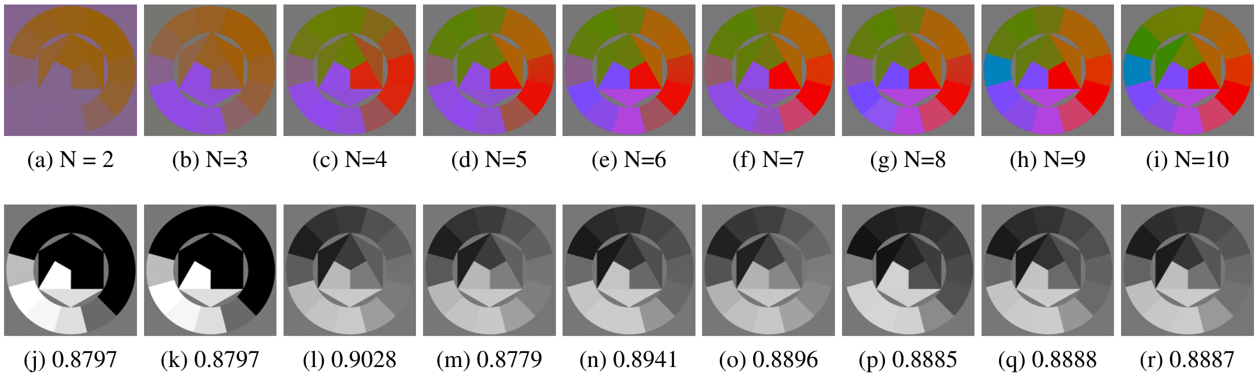
SENSITIVITY STUDY

In this subsection, we test the sensitivity of the proposed method in terms of the parameters, noise, structure and video sequence images.

In our method, there are three parameters. The default parameters are given as follows. In the L_0 smoothing step, λ is the regularization parameter that controls the smoothness of the resulted image. The larger λ is, the smoother the result is. The cluster number is the only parameter in the clustering step. We let N varies from 2 to 10 empirically. Then for each test image, we get nine decolorization result images. To choose the best decolorization result in the nine images, we use the E-score measure. Note that there is a threshold integer τ in the E-score calculation, which denotes the color or grayscale difference of pixel pairs. We set $\tau = 9$ and select the decolorization image with the highest E-score as our final result.

Let us study the sensitivity of our method to the parameters. Firstly, we test the influence of class number N. In Fig. 5, we show the clustering and decolorization results of image Fig. 1a in Cadík's dataset with different N varies from 2 to 10. The results of $N = 2$ and $N = 3$ are similar, where the black regions cannot be distinguished. The results of the others seem similar. By careful observation, we find that the results are quite good for $N = 4,8,10$, which have good contrast and all the distinctive colors have distinct gray values. The E-scores are reported below each gray image. When $N = 4$, the decolorized image has the highest E-score and good visual quality, which is chosen as the final result of the proposed method.

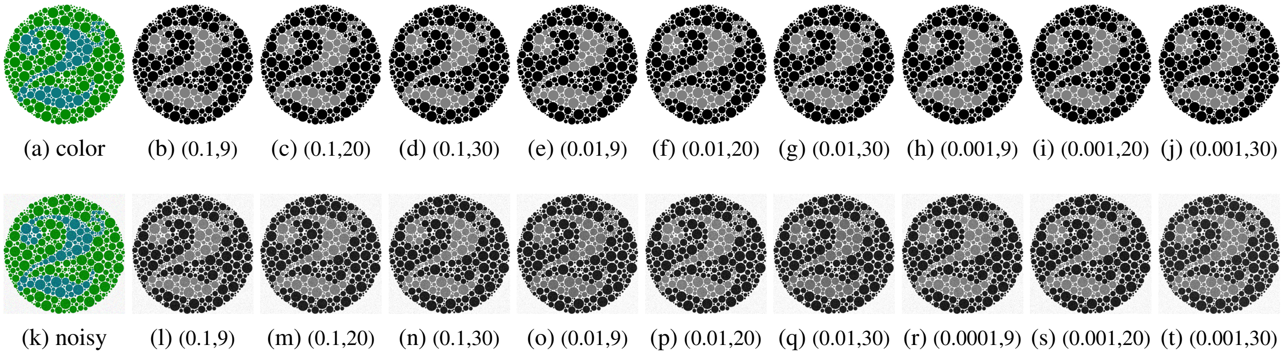
Next, we test the sensitivity of parameters λ, τ and noise in our method. In Fig. 6 and Fig. 7, we test our method on two images in Cadík's dataset with different parameter pairs (λ,μ). We set $\lambda = 0.1,0.01,0.001$ and $\tau = 9,10,20$, such that there are nine pairs of parameters. In Figs. 6-7, the decolorization results in the first rows are the results of the clean color images, while the second rows are the results of the noisy color images which are contaminated by Gaussian noise with zero mean and standard deviation 20. All the decolorization images in Fig. 6 seem quite similar in terms of contrast. In Fig. 7, for the clean color image,



(a) N = 2 (b) N=3 (c) N=4 (d) N=5 (e) N=6 (f) N=7 (g) N=8 (h) N=9 (i) N=10

(j) 0.8797 (k) 0.8797 (l) 0.9028 (m) 0.8779 (n) 0.8941 (o) 0.8896 (p) 0.8885 (q) 0.8888 (r) 0.8887

Fig. 5: Decolorization results of the image in Fig. 1a with different class numbers $N = 2,4,6,8,10$. Each column shows the clustering result and the corresponding decolorized gray image with E-score.



(a) color (b) (0.1,9) (c) (0.1,20) (d) (0.1,30) (e) (0.01,9) (f) (0.01,20) (g) (0.01,30) (h) (0.001,9) (i) (0.001,20) (j) (0.001,30)

(k) noisy (l) (0.1,9) (m) (0.1,20) (n) (0.1,30) (o) (0.01,9) (p) (0.01,20) (q) (0.01,30) (r) (0.0001,9) (s) (0.001,20) (t) (0.001,30)

Fig. 6: Parameter sensitivity of the proposed method. The parameters (λ,τ) are shown below each subfigure. (a) Clean test image "2.png" in Cadik's dataset; (b)-(j) final results for (a) with different parameters; (k) noisy test image which is contaminated by Gaussian noise with zero mean and standard deviation 20; (l)-(t) final results for (k) with different parameters.

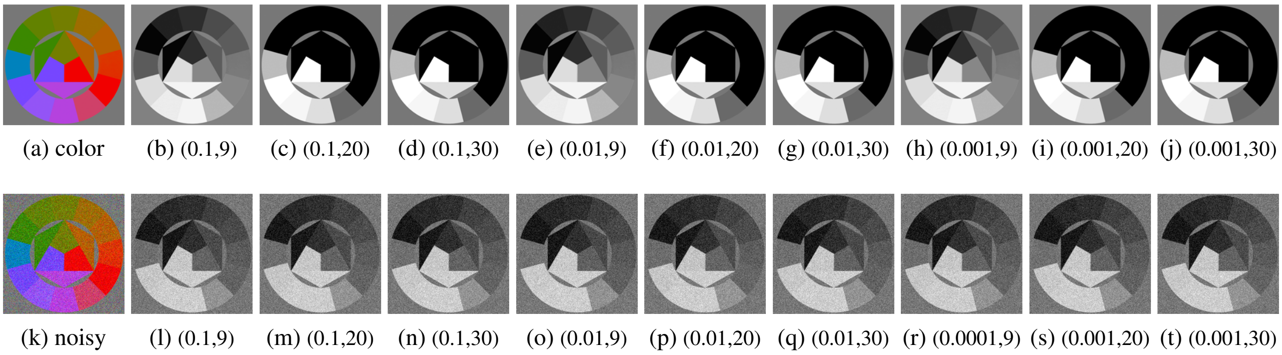
the results are different in contrast and dynamic range. Figs. 7cdfgij are similar and have a high dynamic range; however, the dark regions are not distinctive. While Figs. 7beh have low dynamic range but good region contrast. We also find that for the noisy color image in Fig. 7, our decolorization results are quite robust to both parameters and noise.

In the structure of the proposed method, there are three steps: smoothing, clustering and decolorization. In the first two steps, there are many choices. We test some different combinations on Cadík's dataset. For smoothing, there are four choices: no smoothing (NONE), L_0 smoothing Xu *et al.* (2011), WLS Min *et al.* (2014), SDF Ham *et al.* (2017). For clustering, there are two choices: FCM and FFCM. The computational time and average E-score are reported in Table 1. From Table 1, we find that FCM without smoothing is the slowest; meanwhile, its average E-score is the lowest. Both L_0+FFCM and WLS+FFCM are about four times faster than NONE+FCM. SDF+FFCM is about 2.7 times faster than NONE+FCM. The average E-scores of the decolorization results are quite similar. Among all, L_0+FFCM has a little higher average E-score than others. We remark that the computational time reported in Table 1 is the total computational time of our method which includes all the preprocessing steps and the calculation of the E-score for each candidate decolorized image.

smoothing	clustering	E-score	Time
NONE	FCM	0.9101	41.3s
L_0	FFCM	0.9122	10.1s
WLS	FFCM	0.9106	9.4s
SDF	FFCM	0.9104	15.3s

Table 1: The performance of the proposed decolorization method with different choices of smoothing and clustering methods on Cadík's dataset. The average E-score and average computational time are reported.

Our method is designed for a single image decolor. It has limitations in processing color video sequences



(a) color (b) (0.1,9) (c) (0.1,20) (d) (0.1,30) (e) (0.01,9) (f) (0.01,20) (g) (0.01,30) (h) (0.001,9) (i) (0.001,20) (j) (0.001,30)

(k) noisy (l) (0.1,9) (m) (0.1,20) (n) (0.1,30) (o) (0.01,9) (p) (0.01,20) (q) (0.01,30) (r) (0.0001,9) (s) (0.001,20) (t) (0.001,30)

Fig. 7: Parameter sensitivity of the proposed method. The parameters (λ,τ) are shown below each subfigure. (a) Clean test image; (b)-(j) final results for (a) with different parameters; (k) noisy test image which is contaminated by Gaussian noise with zero mean and standard deviation 20; (l)-(t) final results for (k) with different parameters.

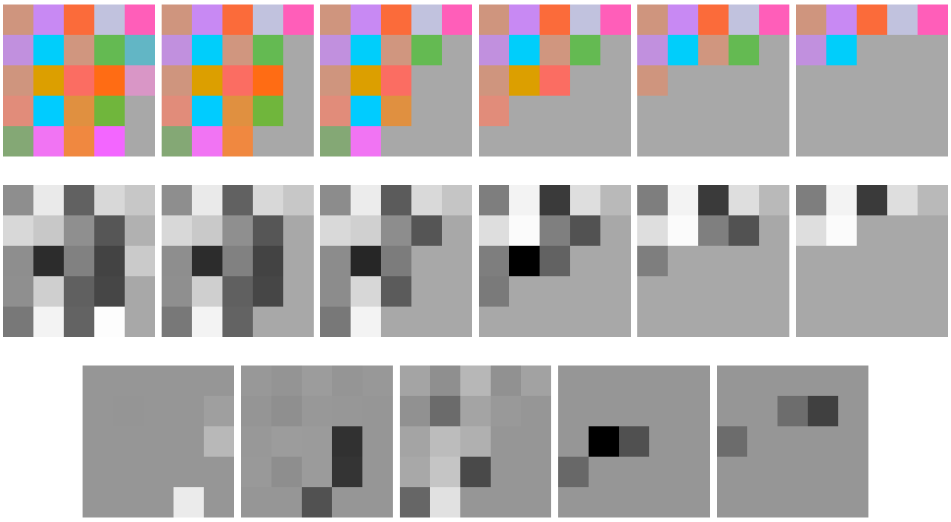
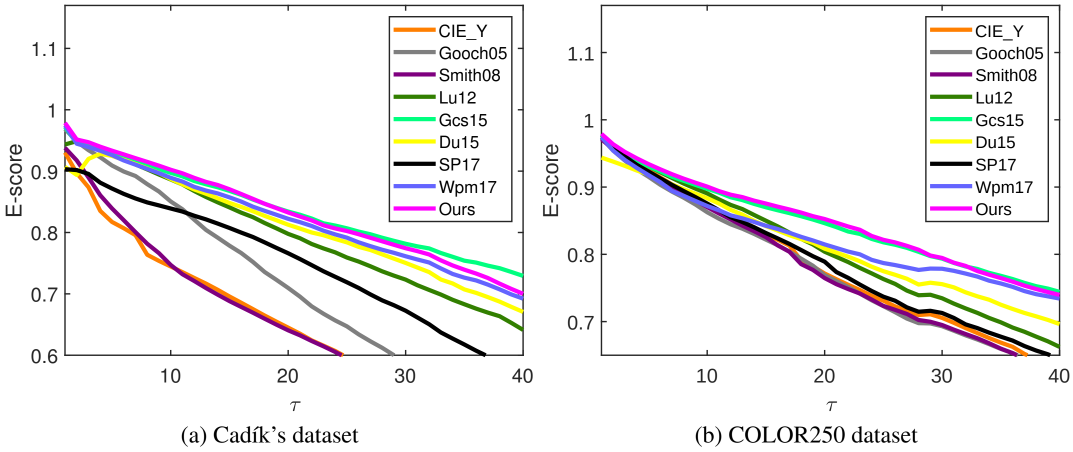


Fig. 8: Video decolorization by the proposed method. First row: the color images in some frames. Second row: our decolorization results. Third row: the difference of two adjacent frames in the second row (adding 150 for better display).

since the relation between frames is not considered. We test a video in Fig. 8 in which the decolorization results of six frames are displayed. The difference images of the decolorized results in the adjacent frames are shown in the last row. From the results, it is easy to see that the first three frames are almost consistent and the last three frames are almost consistent for each color. However, the difference between the third and fourth frames is obvious.

QUANTITATIVE EVALUATION

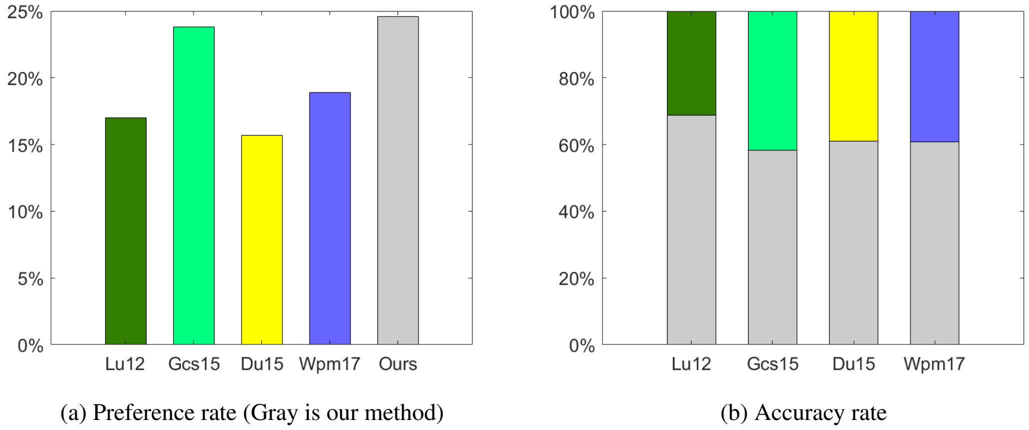
In the following experiments on Cadík's dataset and Color250 dataset, we use the default parameters $\lambda = 0.01, N = 2,...,10, \tau = 9$ if not specified. For quantitative assessment, we compare the E-score measure of the decolorization results of each method. We let the threshold τ in E-score varies from 1 to 40, as suggested by Lu *et al.* (2014). The E-score curves of the two test datasets are shown in Fig. 9a and Fig. 9b, respectively. Higher E-score means better image quality. For Cadík's dataset, when $\tau \in [0,20)$, our method gives the highest E-score. It indicates that our method can preserve the low contrast (the color difference is not very prominent) in the color image better than others. Actually, there are many low contrast regions in color images with abundant colors. When $\tau \geq 20$, Gcs17 has the highest E-score and our method is second best. For COLOR250 dataset, when $\tau \in [0,32]$, our method gains the highest E-score among all, see Fig. 9b. Gcs17 have a very close E-score as ours, which are higher than others.



(a) Cadík's dataset

(b) COLOR250 dataset

Fig. 9: Qualitative evaluation of different methods on two datasets with E-score. The x-axis denotes the threshold τ involved in the calculation of E-score.



(a) Preference rate (Gray is our method)

(b) Accuracy rate

Fig. 10: Results of the preference experiment and accuracy experiment in user study. The gray bar denotes our method. Cadík's dataset is tested.

USER STUDY

We use all the 24 images in Cadík's dataset for the user study. In this study, we invited 20 participants (8 males and 12 females) at the age of 20–41 with no eye-sight deficiency, who are students or teachers in the university. In the experiments, six methods are mainly compared: Cadik08, Lu12, Gcs15, Du15, Wpm17 and ours. For Cadik08, we choose the best results according to the ranking of scores (see Cadík (2008)) involving seven decolorization methods. Note that averagely the last five methods have the top five E-score values as shown in Fig. 9.

The user experiment consists of two parts: preference experiment and accuracy experiment. In the preference experiment, two decolorized images were displayed at two sides of the color image every time. Observers were instructed to select the decolorized image that they preferred. The two images are the results of our method and one of the five methods in Cadik08, Lu12, Gcs15, Du15, and Wpm17. Each participant is asked to choose $24 \times 5 = 120$ image pairs. In the accuracy experiment, every time, five decolorized images obtained by Cadik08, Lu12, Gcs15, Du15, and Wpm17, and our method were displayed along with the color original in the third. Observers were asked to select the decolorized images that best match the original color image in appearance. Results of the user experiment are shown in Fig. 10. We find that the proposed method outperforms the other methods in terms of both preference and accuracy. The preference ratio of our method versus Cadik08, Lu12, Gcs15, Du15, and Wpm17 are 0.556, 0.688, 0.583, 0.609, and 0.608 respectively; see Fig. 10a. All the values are bigger than 0.5, which means that our method is better than others. The preference of Lu12 is the lowest. The accuracy comparison is displayed in Fig. 10b. It's obvious that the ranking from best to worst is: ours, Gcs15, Cadik08, Wpm17, Du15, and Lu12.

QUALITATIVE EVALUATION

For qualitative evaluation, we list some results of Cadík's dataset in Fig. 11 and some results of COLOR250 dataset in Fig. 12. As can be seen from Figs. 11-12, our results in the last column seem quite competitive with the best of the others. Visually, our results have good contrast. Meanwhile, the distinctiveness of colors is well preserved in the decolorized images by our method.

Let's show some details. In Figs. 11, the digital "2" in the first row is much more evident in the results of Wpm17 and ours than others. In the second row, the results of Lu12, Gcs15, Wpm17 and our method are similar and are much better than others. In the third row, the color blocks are distinctive in our results, which outperforms others. In the fourth row, the decolorized white flowers are more visually pleasing than dark flowers. Among the white flowers, our method can preserve the details in the flower and leaves better than others; and Wpm17 can preserve better contrast of the flower and the background than others. For the sunrise image in the last row, the results of Gcs15, Wpm17 and our method have better contrast than others.

In Figs. 12, in the first row, the jacket's color contrast is better preserved by CIE-Y, Gcs15, Du15 and ours. In the second and third rows, the overall contrast of our results seems quite good. In the fourth row, the colors of the three blocks are different; our result is the best among all which preserves the color contrast very well. In the last row, the results of Gcs15 and our methods are better than others since different color blocks are distinctive.

CONCLUSIONS AND FUTURE WORK

We presented a clustering-guided decolorization method in this paper. The clustering step guided us to find distinctive colors. Then we find the optimal linear combination weights by constructing the correspondence of distinctive colors with specified distinctive gray values. We can get many decolorized images by setting different class numbers. We choose the final decolorized image with the best E-score.

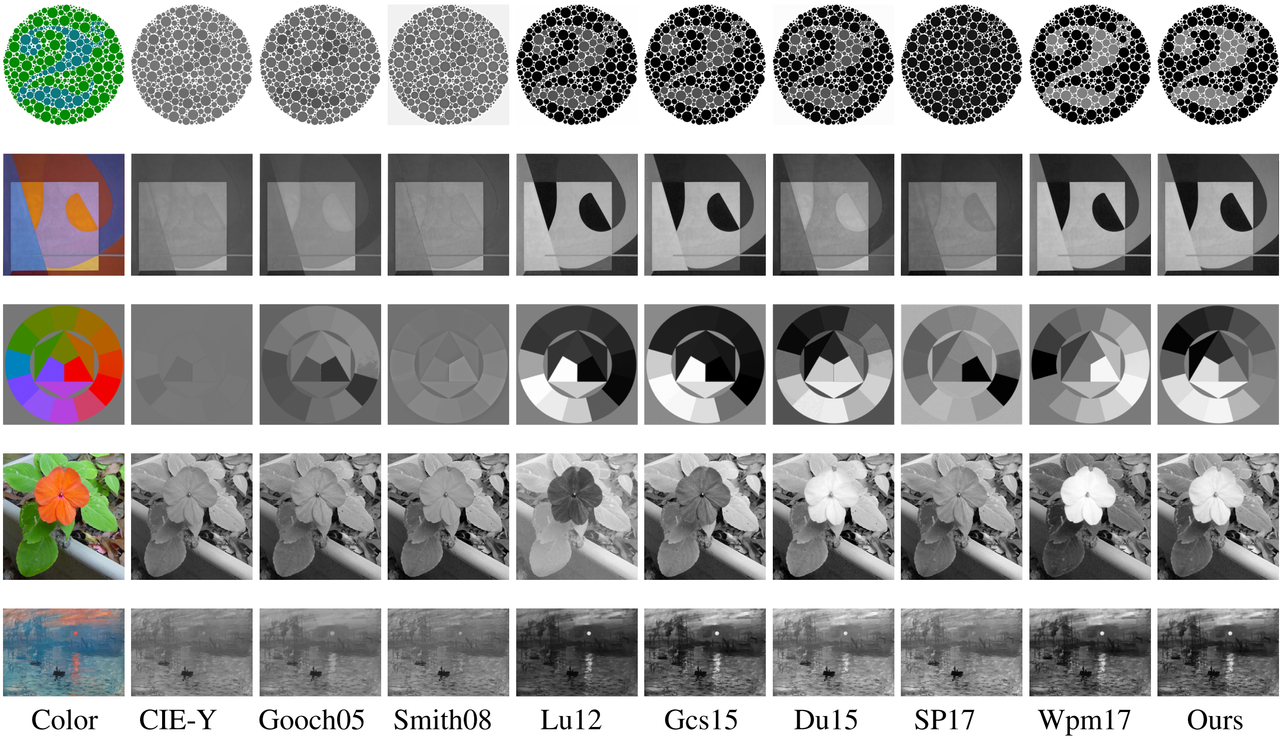


Fig. 11: Decolorization results of different methods on some images in Cadík's dataset. The left column is the original color image. The second to the last columns are results of different methods. The last row show the names of the methods for each column.

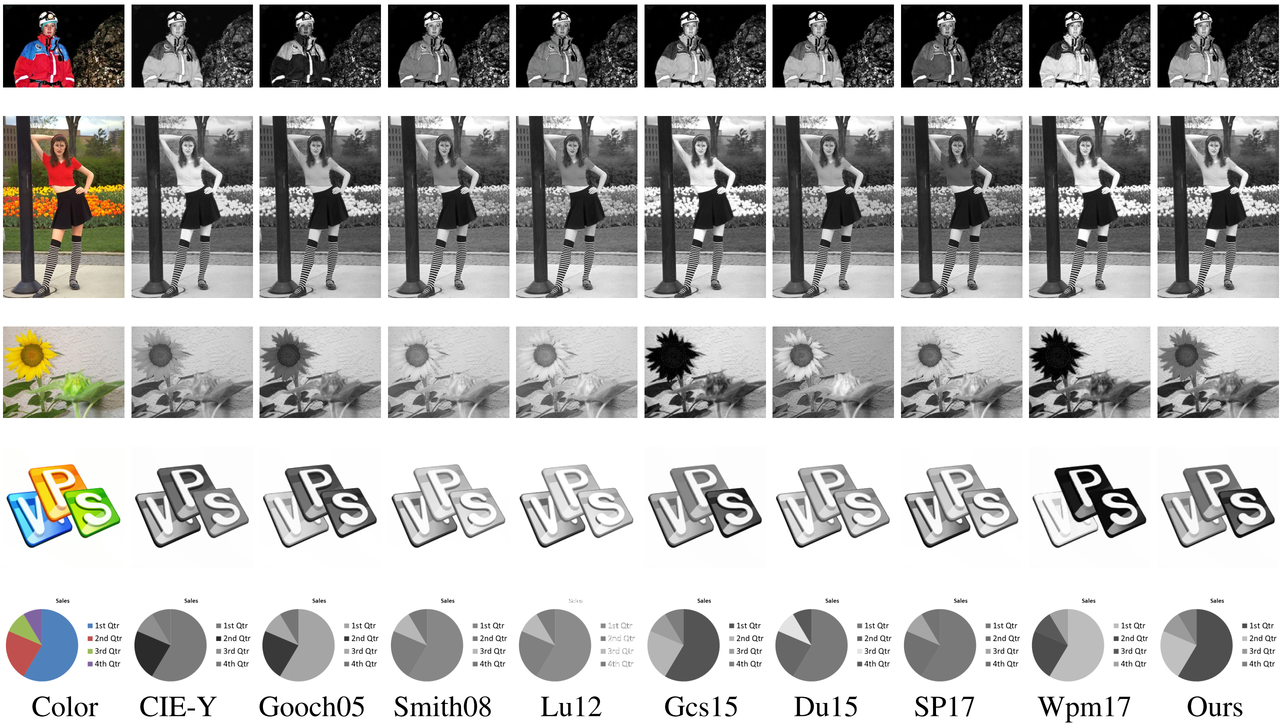


Fig. 12: Decolorization results of different methods on some images in COLOR250 dataset. The left column is the original color image. The second to the last columns are results of different methods. The last row show the names of the methods for each column.

There are several limitations of our method. 1) E-score is not consistent with the human visual system in some tests. So the final result was chosen by E-score maybe not optimal. 2) The proposed method is time-consuming since it has several steps, including smoothing, clustering, optimization, and E-score calculation. It takes about 9 seconds for decolorizing an image with size 390 times 390, in which E-score calculation takes about 3 seconds. 3) The proposed method lacks consistency between frames for video decolorization. Our future work will focus on overcoming the above limitations which include: designing a new measure which is more consistent with the human visual system; reducing the computational time by optimizing all the steps in our method; and studying consistent video decolorization model.

ACKNOWLEDGEMENTS

This work is supported in part by the National Natural Science Foundation of China (NSFC) (No.61731009, No.11671002), the NSFC-RGC under Grant 61961160734, the Fundamental Research Funds for the Central Universities, and Science and Technology Commission of Shanghai Municipality (No.19JC1420102, No.18dz2271000).

REFERENCES

Bala R, Eschbach R (2004). Spatial color-to-grayscale transform preserving chrominance edge information. In: Color and Imaging Conference, vol. 2004. Society for Imaging Science and Technology.

Bertalmio M, Caselles V, Provenzi E, Rizzi A (2007). Perceptual color correction through variational techniques. IEEE T Image Process 16:1058–72.

Bezdek JC, Ehrlich R, Full W (1984). FCM: The fuzzy c-means clustering algorithm. Comput Geosci 10:191–203.

Cadík M (2008). Perceptual evaluation of color-to-grayscale image conversions. In: Comput Graph Forum, vol. 27. Wiley Online Library.

Du H, He S, Sheng B, Ma L, Lau RW (2015). Saliency-guided color-to-gray conversion using region-based optimization. IEEE T Image Process 24:434–43.

Gooch AA, Olsen SC, Tumblin J, Gooch B (2005). Color2gray: salience-preserving color removal. Acm T Graphic 24:634–9.

Grundland M, Dodgson NA (2007). Decolorize: Fast, contrast enhancing, color to grayscale conversion. Pattern Recogn 40:2891–6.

Ham B, Cho M, Ponce J (2017). Robust guided image filtering using nonconvex potentials. IEEE T Pattern Anal 40:192–207.

Hou X, Duan J, Qiu G (2017). Deep feature consistent deep image transformations: Downscaling, decolorization and HDR tone mapping. arXiv preprint arXiv170709482 .

Ji Z, Fang Me, Wang Y, Ma W (2016). Efficient decolorization preserving dominant distinctions. The Visual Computer 32:1621–31.

Jin Z, Li F, Ng MK (2014). A variational approach for image decolorization by variance maximization. Siam J Imaging Sci 7:944–68.

Kim Y, Jang C, Demouth J, Lee S (2009). Robust color-to-gray via nonlinear global mapping. ACM T Graphic 28:1–4.

Lau C, Heidrich W, Mantiuk R (2012). Cluster-based color space optimizations. In: IEEE I Conf Comp Vis.

Liu Q, Leung H (2018). Variable augmented neural network for decolorization and multi-exposure fusion. Inform Fusion 46:114–27.

Liu Q, Li S, Xiong J, Qin B (2019). WpmDecolor: weighted projection maximum solver for contrast-preserving decolorization. Visual Comput 35:205–21.

Liu Q, Liu P, Wang Y, Leung H (2017a). Semi-parametric decolorization with laplacian-based perceptual quality metric. IEEE T Circ Syst Vid 27:1856–68.

Liu Q, Liu PX, Xie W, Wang Y, Liang D (2015). GcsDecolor: Gradient correlation similarity for efficient contrast preserving decolorization. IEEE T Image Process 24:2889–904.

Liu Q, Shao G, Wang Y, Gao J, Leung H (2017b). Log-euclidean metrics for contrast preserving decolorization. IEEE T Image Process 26:5772–83.

Liu S, Zhang X (2019). Image decolorization combining local features and exposure features. IEEE T Multimedia 21:2461–72.

Lu C, Xu L, Jia J (2012a). Contrast preserving decolorization. In: IEEE INT Conf Comput. IEEE.

Lu C, Xu L, Jia J (2012b). Real-time contrast preserving decolorization. In: SIGGRAPH Asia 2012 Technical Briefs. ACM.

Lu C, Xu L, Jia J (2014). Contrast preserving decolorization with perception-based quality metrics. INT J Comput Vision 110:222–39.

Min D, Choi S, Lu J, Ham B, Sohn K, Do MN (2014). Fast global image smoothing based on weighted least squares. IEEE T Image Process 23:5638–53.

Neumann L, Čadík M, Nemcsics A (2007). An efficient perception-based adaptive color to gray transformation. In: Proceedings of the Third Eurographics conference on Computational Aesthetics in Graphics, Visualization and Imaging. Eurographics Association.

Rasche K, Geist R, Westall J (2005). Detail preserving reproduction of color images for monochromats and dichromats. IEEE Comput Graph 25:22–30.

Smith K, Landes PE, Thollot J, Myszkowski K (2008). Apparent greyscale: A simple and fast conversion to perceptually accurate images and video. In: Comput Graph Forum, vol. 27. Wiley Online Library.

Szilagyi L, Benyo Z, Szilagyi SM, Adam HS (2003). MR brain image segmentation using an enhanced fuzzy c-means algorithm. In: P Ann Int IEEE EMBS, vol. 1.

Wang W, Li Z, Wu S (2018). Color contrast-preserving decolorization. IEEE T Image Process 27:5464–74.

Xu L, Lu C, Xu Y, Jia J (2011). Image smoothing via L0 gradient minimization. In: ACM T Graphic, vol. 30. ACM.

Zhang X, Liu S (2018). Contrast preserving image decolorization combining global features and local semantic features. The Visual Computer 34:1099–108.

Zhao H, Zhang H, Jin X (2018). Efficient image decolorization with a multimodal contrast-preserving measure. Comput Graph 70:251–60.