

Instant Edit Propagation on Images Based on Bilateral Grid

Feng Li^{1, 2}, Chaofeng Ou², Yan Gui^{1, 2, 3, *} and Lingyun Xiang^{1, 2}

Abstract: The ability to quickly and intuitively edit digital content has become increasingly important in our everyday life. However, existing edit propagation methods for editing digital images are typically based on optimization with high computational cost for large inputs. Moreover, existing edit propagation methods are generally inefficient and highly time-consuming. Accordingly, to improve edit efficiency, this paper proposes a novel edit propagation method using a bilateral grid, which can achieve instant propagation of sparse image edits. Firstly, given an input image with user interactions, we resample each of its pixels into a regularly sampled bilateral grid, which facilitates efficient mapping from an image to the bilateral space. As a result, all pixels with the same feature information (color, coordinates) are clustered to the same grid, which can achieve the goal of reducing both the amount of image data processing and the cost of calculation. We then reformulate the propagation as a function of the interpolation problem in bilateral space, which is solved very efficiently using radial basis functions. Experimental results show that our method improves the efficiency of color editing, making it faster than existing edit approaches, and results in excellent edited images with high quality.

Keywords: Instant edit propagation, bilateral grid, radial basis function, image editing.

1 Introduction

With the development of information technology, digital media technology [Shen, Shen, Liu et al. (2018); Xiang, Shen, Qin et al. (2018), Shen, Shen, Sun et al. (2018)] has become widely used, and people accordingly have higher requirements for visual media quality. Thanks to powerful image editing tools, people without any knowledge about image processing can easily edit a photo in a visually plausible way. The edit processing of images/videos has become a research hotspot, and color processing technology is an important part of this field. Color processing technology mainly includes color transfer and color editing propagation [He, Gui and Li (2017)].

¹ Hunan Provincial Key Laboratory of Intelligent Processing of Big Data on Transportation, Changsha University of Science and Technology, Changsha, 410114, China.

² School of Computer and Communication Engineering, Changsha University of Science and Technology, Changsha, 410114, China.

³ School of Mechanical and Aerospace Engineering, Nanyang Technological University, Singapore, 639798, Singapore.

* Corresponding Author: Yan Gui. Email: guiyan122@163.com.

Color transfer technology enables the mapping of color information from a reference image to a target image, thereby changing the visual effect of the target image [Faridul, Pouli, Chamaret et al. (2016); Irony, Cohen-Or and Lischinski (2005); Reinhard, Adhikhmin, Gooch et al. (2002); Welsh, Ashikhmin and Mueller (2002)]. However, it is difficult to choose an appropriate reference image for transferring color. In order to solve the problem of reference image selection [Xie, Qin, Xiang et al. (2018)], machine learning [Xiang, Zhao, Li et al. (2018)] can be applied to color transfer [Cheng, Yang and Sheng (2015); Iizuka, Simo-Serra and Ishikawa (2016)]. Using machine learning methods, an appropriate reference image can be learned by training a prepared reference image set, which should always contain a large number of different of reference images types. Machine-learning-based color transfer methods can reduce the level of required user interaction and the complexity of selecting reference images. However, when color processing is applied to complex natural images, the color consistency and spatial continuity of the repeated scene elements cannot be maintained during the color transfer process.

By contrast, editing propagation preserves the color consistency and spatial continuity of repeated scene elements. Color editing propagation first requires the user to draw colored lines on the image, then uses algorithms to achieve color transfer from the marked position to the unknown color area. Compared with color transfer, this method has the advantage of enabling people to mark different parts of the image with different colors according to requirements, meaning that the image will satisfy the color requirement to a greater content. Grayscale image coloring and color image re-coloring are two important methods of editing propagation [He, Gui and Li (2017)].

The grayscale image colorization method based on user interaction is an important part of color editing propagation [Horiuchi and Hirano [2003]; Horiuchi and Kotera [2006]; Levin, Lischinski and Weiss [2004]]. This method requires numerous interactions, performs gray value similarity determination for each pixel in the image, and edits each pixel point in subsequent edit propagation; as a result, it occupies a large amount of memory and costs a lot of time.

Compared with the grayscale image colorization method, the color image re-coloring method requires only user-provided color scribbles of the local image area, meaning that the user operation is simpler [Xu, Li, Ju et al. (2009); Pellacini and Lawrence (2007); Huang, Zhang and Martin (2015)] without requiring to extract the local features of the image [Li, Qin, Xiang et al. (2018)]. However, if the image size is large, the storage memory required will increase; this occupation of the large amount of memory reduces computational efficiency, resulting in longer re-coloring times.

Existing color processing methods usually take tens or even hundreds of seconds to process a picture. These methods are inefficient and highly time-consuming. In order to solve the problem of low time efficiency, we propose an instant edit propagation method based on a bilateral grid. Our method is divided into two phases: the image preprocessing phase and the editing propagation phase. In the image preprocessing phase, we resample the input images via interactions using a bilateral grid. The editing propagation phase involves propagating the color feature information of the grid vertices of the interactive marks to other grid vertices according to the interpolation model. Experimental results

show that our method can greatly improve the speed of color editing and yields high-quality editing results.

The Section 2 mainly describes some works related to this paper. The Section 3 will present the framework of our method. The Section 4 is mainly concerned with the presentation of experimental results and drawing comparisons with existing editing propagation methods. Finally, the Section 5 presents the conclusion.

2 Related work

Our work is inspired by earlier work on color transfer, but mainly uses edit propagation.

Color transfer. Color transfer involves mapping the color distribution of a reference image to the input image [Tai, Jia, Tang et al. (2005); Pitié, Kokaram and Dahyot (2007)]. Xiao et al. [Xiao, Wan, Leung et al. (2015)] proposed a color transfer method based on gradient mesh optimization that can effectively maintain the gradient network structure information of images in color transfer results. On this basis, Liu et al. [Liu and Song (2018)] proposed a multi-source image color migration method based on edit propagation. This method proposes an interactive, automatic correction method that uses the modified interactive mark as the control sample in the edit propagation, finds all regions in the image similar to the interactive mark, and combines color migration and apparent migration technology, such that the corresponding color or texture information is migrated to the target image in order to obtain the migration result. Accordingly, our work draws on the advantages of both approaches: We optimize the interpolation model to accurately propagate the interactive color information to the grid vertices.

Edit propagation. User interaction-based methods can help users to propagate editing operations from one image region to another region with a similar appearance [An, and Pellacini (2008); Bie, Huang and Wang (2011)]. Li et al. [Li, Ju and Hu (2010)] proposed a fast image editing method based on the radial basis function interpolation model, which can edit other image regions that are not interactive but are similar to the sample points. To solve color mixing in Li's method, Chen et al. [Chen, Chen and Zhao (2012)] employed local linear embedding (LLE); however, when the value of the K-domain [Xiao, Wang and Liu (2018)] is large, the expected color conversion effect cannot be achieved. This method solves the data overflow presenting in the first two methods. Xu et al. [Xu, Yan and Jia (2013)] proposed an image editing method based on the sparse control model, which can achieve better color editing results with a small amount of user interaction. Combining the technical characteristics of Li et al. [Li, Ju and Hu (2010)], Chen et al. [Chen, Chen and Zhao (2012)] and Xu et al. [Xu, Yan and Jia (2013)], we propose an instant image editing method based on the bilateral grid. The method only requires a small amount of interaction to ensure the quality and instant editing of the image.

Recently, Chen et al. [Chen, Li, Chen et al. (2016)] proposed learning sparse dictionaries for edit propagation. This method follows the principle of sparse representation to obtain a representative and compact dictionary and perform edit propagation on the dictionary instead; accordingly, it reduces memory consumption while still maintaining a high degree of visual fidelity.

3 The proposed method

3.1 Framework

In order to improve the speed of image color editing, we propose an interactive color editing method, a flowchart of which is shown in Fig. 1. The method includes two main stages: image preprocessing and color editing propagation.

(1) **Image preprocessing.** In this stage, the user needs to perform color line marking on the local area. However, when the grayscale channel of the color of interactive input is very different from the grayscale channel of the original image, the color will exhibit a large deviation after coloring is complete. Accordingly, we adjust the grayscale channels to accommodate the grayscale channels of interactive colors. Finally, the bilateral grid is used to resample the input image via user interaction.

(2) **Color editing propagation.** This stage consists of two parts: the construction of the interpolation model and the color editing propagation. When constructing the interpolation model, the grid vertex's feature vector of the original image grid is used as input, while the feature vector of the user edits grid is used for interpolation to construct the radial basis interpolation model. During color editing propagation, the color values of the marked grid vertex are propagated to the target grid, after which the color values of the grid vertex are inversely mapped to individual pixels within the grid vertex. Finally, all pixels in the input image grid are inversely mapped to the *RGB* color space to obtain the output image after re-coloring.

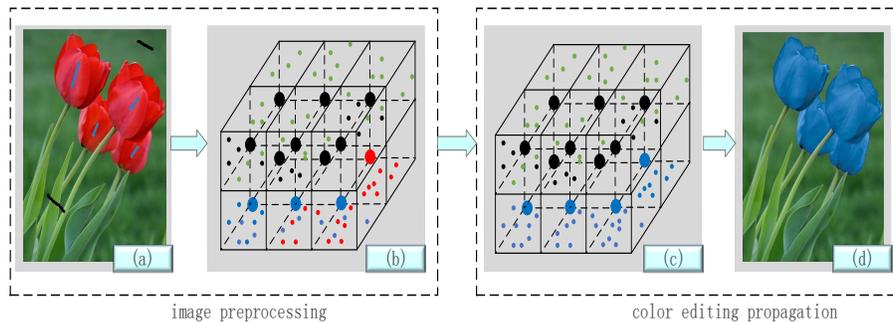


Figure 1: Method flowchart: (a) the input image; (b) the bilateral grid after image preprocessing; (c) the bilateral grid after color editing; (d) the output image

The advantages of our method are that it uses the grid vertex to replace the image pixel, significantly reducing the amount of image data to be edited, and can greatly improve the speed of color editing.

3.2 Image preprocessing

To improve time efficiency, we resample input images with interactions by using a bilateral grid [Chen, Paris and Durand (2007)]. The bilateral grid can be described as a five-dimensional array where the first three dimensions represent the *Lab* value and the last two dimensions correspond to the image space coordinates. As a result, the input image data can be described in the high-dimensional feature space.

It should be noted that all experiments in this paper were performed in the *Lab* space. The related parameters s_r , s_c and s_s are respectively the sampling rate of the gray value axis, *ab* channel, and the spatial coordinate axis (x, y) . Intuitively, the size of the bilateral grid is inversely proportional to the sampling rate of each dimension. By setting the appropriate sampling rate, the number of bilateral grid vertices will be much smaller than the total number of pixels in the image, which can greatly reduce the amount of image data needing to be edited during color editing propagation.

In this paper, given an input image I with user interaction, in a bilateral grid of a specified size, each image pixel will be mapped to the corresponding grid cell according to the following formula:

$$\Gamma([c_r / s_r], [c_a / s_a], [c_b / s_b], [x / s_x], [y / s_y])_+ = (I(x, y), 1) \quad (1)$$

where $[\cdot]$ is the closest-integer operator for calculating a grid vertex coordinate corresponding to each pixel. The homogeneous coordinates $(I(x, y), 1)$ are used to accumulate the color values and the total number of pixels in each grid cell. In order to define the subsequent bilateral grid-based editing propagation, we also need to obtain the color values C_a^i of each grid vertex v_i :

$$C_a^i = \sum_{j=1}^{N_i} (c_a)_i^j / N_i \quad (2)$$

where $(c_a)_i^j$ is the color value of the j^{th} unmarked pixel in the grid vertex v_i , while N_i is the total number of unmarked pixels in the grid cell. We next need to calculate the coordinate values of the grid vertex. However, if we use the average coordinate value of the pixel points contained in the grid to define the grid vertex coordinates, a coordinate value overflow may occur. In order to solve this overflow, the coordinate value X_i of the grid vertex v_i is defined as the coordinate value of the last pixel that enters the grid.

Since the color values of all marked pixel points of the same color are the same, the color value $C_a^{j'}$ of the marked grid vertex v_j is defined herein as the color value of the marked pixel point.

Thus, the input feature vector for each grid vertex v_i is represented as $\mathbf{f}_i = (C_a^i, C_b^i, X_i, Y_i)$, while the interactive feature vector for each marked grid vertex v_j is represented as $\mathbf{g}_j = (C_a^{j'}, C_b^{j'}, X_j, Y_j)$. Since the feature vector of the grid vertex does not contain gray values, the above two feature vectors are four-dimensional.

It should be noted that the grid cells containing black marker pixels do not require calculation of the color values and coordinate values of their vertex; these grid cells will also be ignored in the subsequent re-coloring of the grid vertex.

3.3 Color editing propagation

In the bilateral space, we transform the editing propagation into a discrete point interpolation problem. We define the total set of all marked grid vertices as G .

In this paper, we use the least squares method to define the energy function [Amiri-Simkooei, Mortazavi and Asgari (2015)]. Constructing the energy function in this way allows for minimization of the editing difference between the vertex of the re-coloring grid and the vertex of the marked grid after the energy function is minimized. The energy function is as follows:

$$\mathbf{J} = \sum_{i \in G} (\mathbf{g}_{v_i} - \mathbf{h}(\mathbf{f}_{v_i}))^2 \quad (3)$$

where \mathbf{g}_{v_i} is the marked feature vector corresponding to the grid vertex v_i , while \mathbf{f}_{v_i} is an input feature vector of the grid vertex v_i . $\mathbf{h}(\mathbf{f}_{v_i})$ is the feature vector edited by the grid vertex v_i . This paper defines $\mathbf{h}(\mathbf{f})$ as the radial basis function [Wang (2017)]:

$$\mathbf{h}(\mathbf{f}) = \sum_{i \in G} \mathbf{a}_{v_i} \phi(\|\mathbf{f} - \mathbf{f}_{v_i}\|) \quad (4)$$

where \mathbf{a}_{v_i} is the weight coefficient of the proportion of the control grid vertex v_i in $\mathbf{h}(\mathbf{f})$, which is unknown. In the subsequent optimization edit, we will explain how to solve \mathbf{a}_{v_i} . \mathbf{f} is the feature vector of the vertex of the grid to be edited, and the radial basis function is $\phi(\mathbf{r}) = \exp(-\mathbf{r}^2)$.

Minimizing the energy function is actually a problem of solving a series of linear equations. The classical method for solving a series of linear equations is to use the normal equation [Huang and Chen (2002)].

Substituting Eq. (3) into Eq. (4) obtains a complete energy function J :

$$\mathbf{J} = \sum_{i \in G} (\mathbf{g}_{v_i} - \sum_{j \in G} \mathbf{a}_{v_j} \phi(\|\mathbf{f}_{v_i} - \mathbf{f}_{v_j}\|))^2 \quad (5)$$

In Eq. (5), the only unknown is the weighting factor, which is obtained by substituting the matrix:

$$(\mathbf{X}\mathbf{a} - \mathbf{Y})^T (\mathbf{X}\mathbf{a} - \mathbf{Y}) = \sum_{i \in G} (\mathbf{h}(\mathbf{f}_{v_i}) - \mathbf{g}_{v_i})^2 = \mathbf{J}(\mathbf{a}) \quad (6)$$

Where \mathbf{X} is the input matrix, \mathbf{a} is the coefficient matrix to be solved, and \mathbf{Y} is the output matrix, $G = [i1, i2, \dots, ii]$.

$$\mathbf{X} = \begin{bmatrix} \sum_{j \in G} e^{-\|\mathbf{f}_{i1} - \mathbf{f}_j\|^2} \\ \sum_{j \in G} e^{-\|\mathbf{f}_{i2} - \mathbf{f}_j\|^2} \\ \vdots \\ \sum_{j \in G} e^{-\|\mathbf{f}_{ii} - \mathbf{f}_j\|^2} \end{bmatrix} \quad (7)$$

$$\mathbf{Y} = \begin{bmatrix} \mathbf{g}_{i1} \\ \mathbf{g}_{i2} \\ \vdots \\ \mathbf{g}_{ii} \end{bmatrix} \quad (8)$$

The above energy formula is obtained by partial derivative of the coefficient matrix and calculated by matrix derivation:

$$\begin{aligned}
 \nabla_{\mathbf{a}}\mathbf{J}(\mathbf{a}) &= \nabla_{\mathbf{a}}(\mathbf{X}\mathbf{a} - \mathbf{Y})^T(\mathbf{X}\mathbf{a} - \mathbf{Y}) \\
 &= \nabla_{\mathbf{a}}(\mathbf{a}^T\mathbf{X}^T\mathbf{X}\mathbf{a} - \mathbf{a}^T\mathbf{X}^T\mathbf{Y} - \mathbf{Y}^T\mathbf{X}\mathbf{a} - \mathbf{Y}^T\mathbf{Y}) \\
 &= \nabla_{\text{atr}}(\mathbf{a}^T\mathbf{X}^T\mathbf{X}\mathbf{a} - \mathbf{a}^T\mathbf{X}^T\mathbf{Y} - \mathbf{Y}^T\mathbf{X}\mathbf{a} - \mathbf{Y}^T\mathbf{Y}) \\
 &= 2(\mathbf{X}^T\mathbf{X}\mathbf{a} - \mathbf{X}^T\mathbf{Y})
 \end{aligned} \tag{9}$$

Let the derivative $\nabla_{\mathbf{a}}\mathbf{J}(\mathbf{a}) = \mathbf{0}$ be expressed as the normal equations: $\mathbf{X}^T\mathbf{X}\mathbf{a} = \mathbf{X}^T\mathbf{Y}$. Therefore, the minimum value of \mathbf{a} is: $\mathbf{a}_{\min} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y}$.

Once the relationship function h is determined, the grid vertex can be re-colored. After re-coloring the grid vertex, the next step is inverting the color of the grid vertex and the vertex of the marked grid to each pixel point it contains, so that the color value of the pixel in the grid is consistent with the color value of the grid vertex. However, the gray value remains unchanged. Finally, all pixels in the input image grid are inversely mapped to the *RGB* color space to obtain the output image after re-coloring.

4 Experimental results

4.1 Experimental settings

We test on the Windows 7 operating system, which runs on a PC equipped with a 2.80 GHz CPU and 8 GB of memory. Firstly, the optimal sampling rate parameters are obtained, after which the effectiveness of the proposed method is verified by the quality and efficiency of the experimental results.

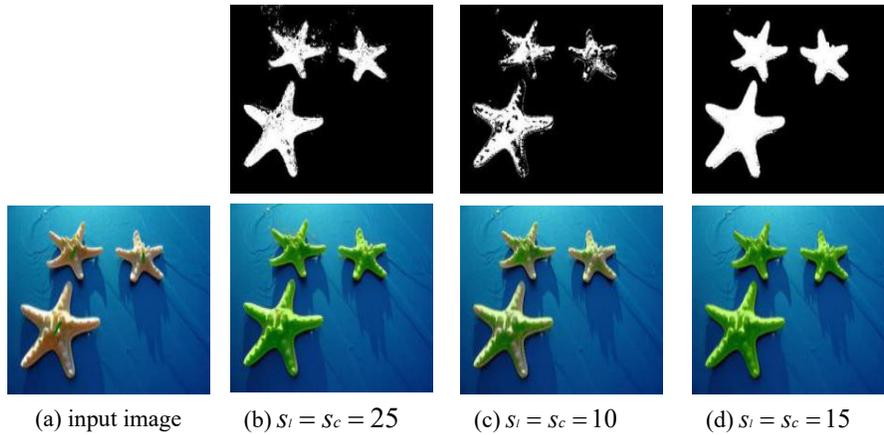


Figure 2: Color editing results at different sampling rates. The figures on the bottom row show the color editing results with different parameters, while the figures on the top show the corresponding probability map

The division of the bilateral grid is determined by the sampling rate. The number of grid cells greatly affects both color editing results and time efficiency. Generally speaking, we have a fixed coordinate sampling rate s_s that is one-tenth of the image size, while the

sampling rates of the gray value axis and ab channel (s_r and s_c) are set to 15. As shown in Fig. 2(c), once the sampling rate is lower, the excessive bilateral grid cells will leave most pixels out of a grid. The number of bilateral grid cells is reduced as the sampling rate increases, which facilitates subsequent color editing and propagation. However, a high sampling rate will result in fewer grids. Once a grid contains a tremendous number of pixels, this will cause a numerical overflow, as shown in Fig. 2(b). After repeated adjustment of the parameters, we select the $s_r = s_c = 15$ sampling rate as the subsequent experimental parameters.

We selected four images randomly downloaded from the Internet as the experimental images, which have been named *starfish*, *fruit*, *children* and *flower*. All of these are natural images that can generally be divided into two classes: one class contains *starfish* and *fruit*, which are the images with repeated scene elements, while the other contains *children* and *flower*, which are the images with complex scene elements.

To conduct the comparison, we employed four different color editing methods to be compared with our method [Li, Ju and Hu (2010); Chen, Chen and Zhao (2012); Xu, Yan and Jia (2013); Chen, Li, Chen et al. (2016)], which are referred to as Li et al. [Li, Ju and Hu (2010); Chen, Chen and Zhao (2012); Xu, Yan and Jia (2013); Chen, Li, Chen et al. (2016)] respectively (because Chen has proposed two different methods, we distinguish between them using the year of publication). In order to get the color edit output image, each method needs to interact with the input image before color editing takes place. Tab. 1 presents the user interaction statistics of the images for the different methods. The interacted images are then color edited, and the results of this process are presented in Fig. 3.

Table 1: User interaction statistics

| Image | Image size /pixel | Number of interactions | | | | |
|----------|----------------------|------------------------|-------------------------|----------------|-------------------------|---------------|
| | | Li's method | Chen's method (2012) | Xu's method | Chen's method (2016) | Our method |
| starfish | 420×315 | 4 | 4 | 6 | 4 | 4 |
| fruit | 400×249 | 6 | 6 | 7 | 3 | 3 |
| child | 320×400 | 4 | 8 | 8 | 4 | 4 |
| flower | 300×400 | 6 | 6 | 6 | 2 | 2 |

Chen et al.'s [Chen, Chen and Zhao (2012)] requires fewer user interactions for a single-color image. However, when color editing is performed on an image with a large color difference, the number of user interactions increases. The advantage of Xu et al.'s [Xu, Yan and Jia (2013)] method is that the color editing propagation takes the spatial positional features into account, enabling variable color editing of the same area. However, its shortcoming is also obvious in that the number of interactions is slightly increased. For both Chen et al.'s [Chen, Li, Chen et al (2016)] method and our method require few user interactions, thus we make the same number of interactions on the same images using these two methods, while making more interactions by Chen et al.'s [Chen, Chen and Zhao (2012); Xu, Yan and Jia (2013)] method.

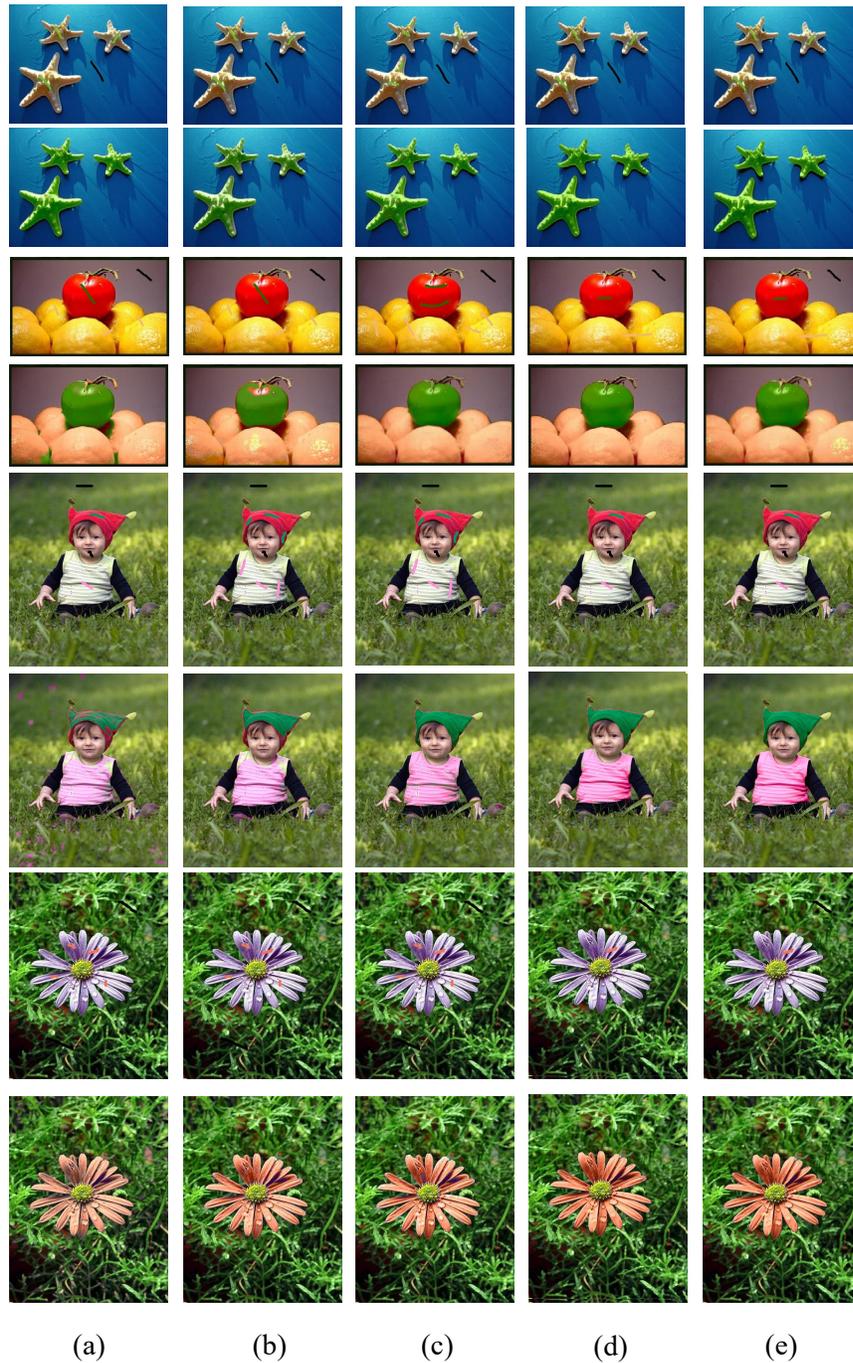


Figure 3: The color editing results of different color editing methods: (a) Li's method; (b) Chen's method (2012); (c) Xu's method; (d) Chen's method (2016); (e) our method. The figures in the odd-numbered line are the input interacted images, while the figures below are the corresponding color editing images

In the subsequent method quality comparison, the quality of the color editing results was quantitatively evaluated using the image's structural similarity index, or SSIM [Wang and Bovik (2002); Wang, Bovik, Sheikh et al. (2004)]. For the original target image I_1 and the color edited image I_2 , the SSIM can be obtained using the following formula:

$$SSIM(I_1, I_2) = \frac{(2u_{I_1}u_{I_2} + c_1)(2\sigma_{I_1}\sigma_{I_2} + c_2)}{(u_{I_1}^2 + u_{I_2}^2 + c_1)(\sigma_{I_1}^2 + \sigma_{I_2}^2 + c_2)} \quad (10)$$

where u_{I_1} and u_{I_2} are the average color values of the image I_1 and the image I_2 respectively. σ_{I_1} and σ_{I_2} are the color variances of the image I_1 and the image I_2 respectively. c_1 and c_2 are the constants added to prevent the denominator from being zero.

4.2 Image quality comparison

Fig. 3 shows the color editing results of our method, Li's method, Chen et al.'s [Chen, Chen and Zhao (2012)] method, Xu et al.'s [Xu, Yan and Jia (2013)] method and Chen et al.'s [Chen, Li, Chen et al. (2016)] method for images with repeated scene elements and complex scene elements. For the color editing images in Fig. 3, we calculated their corresponding SSIM, which are presented in Tab. 2.

Table 2: Statistics of SSIM values

| Image | Image size /pixel | SSIM | | | | |
|---------------|----------------------|----------------|-------------------------|----------------|-------------------------|---------------|
| | | Li's method | Chen's method (2012) | Xu's method | Chen's method (2016) | Our method |
| starfish | 420×315 | 0.753 | 0.778 | 0.825 | 0.834 | 0.850 |
| fruit | 400×249 | 0.736 | 0.769 | 0.805 | 0.814 | 0.822 |
| child | 320×400 | 0.720 | 0.757 | 0.792 | 0.810 | 0.815 |
| flower | 300×400 | 0.724 | 0.762 | 0.814 | 0.821 | 0.831 |
| Average value | | 0.733 | 0.767 | 0.809 | 0.820 | 0.831 |

As shown in Tab. 2, when compared with Li et al.'s [Li, Ju and Hu (2010); Chen, Chen and Zhao (2012); Xu, Yan and Jia (2013)] method, our method has the highest SSIM values for all images with both repeated scene elements and complex scene elements. Li's method results in color mixing and its SSIM values are the lowest. The Chen et al.'s [Chen, Chen and Zhao (2012)] method uses the local linear embedding method to effectively eliminate the boundary color mixing ring and make up for the shortcomings of the Li et al.'s [Li, Ju and Hu (2010)] method; as a result, the SSIM values are higher for Chen's method (2012) than Li et al.'s [Li, Ju and Hu (2010)] method. However, Chen et al.'s [Chen, Chen and Zhao (2012)] method results in color loss when editing colors locate in multiple different areas. Xu et al.'s [Xu, Yan and Jia (2013)] method considers spatial position features to enable variable color editing of the same area more efficiently. Moreover, Xu et al.'s [Xu, Yan and Jia (2013)] method overcomes the overflow and color loss exhibited by Chen et al.'s [Chen, Chen and Zhao (2012)] method, meaning that it has higher SSIM values than Chen et al.'s [Chen, Chen and Zhao (2012)] method,

although it also requires more interactions. For its part, Chen et al.'s [Chen, Li, Chen et al. (2016)] method obtains a representative and compact dictionary and performs edit propagation on the dictionary instead. While Chen et al.'s [Chen, Li, Chen et al. (2016)] method has excellent SSIM values, its time efficiency could still be improved. Finally, our method uses the optimized radial basis interpolation model to greatly improve the editing quality; as a result, our method can generate good-quality new color editing images with the highest SSIM values.

4.3 Image color editing efficiency comparison

Color editing methods usually contain two stages: (1) importing image data and building a model; (2) color editing propagation. We calculated the running time of the two stages of each method in Fig. 3. The results are listed in Tab. 3.

Tab. 3 compares the time taken to conduct image color editing for the images in Fig. 3.

Table 3: Running time statistics

| Image | Image size /pixel | Editing time/s | | | | |
|---------------|----------------------|----------------|-------------------------|----------------|-------------------------|-------------------|
| | | Li's method | Chen (2012)'s method | Xu's method | Chen (2016)'s method | Our method |
| | | Stage 1/2 | Stage 1/2 | Stage 1/2 | Stage 1/2 | Stage 1/2 |
| starfish | 420×315 | 1.35/25.50 | 1.42/25.66 | 2.56/30.72 | 1.68/10.47 | 5.26/ 2.25 |
| fruit | 400×249 | 0.69/10.24 | 0.74/10.55 | 0.98/14.16 | 1.37/5.75 | 2.94/ 1.83 |
| child | 320×400 | 0.96/10.45 | 1.04/15.66 | 1.36/18.79 | 1.07/12.38 | 2.4/ 4.02 |
| flower | 300×400 | 0.87/5.58 | 1.12/6.86 | 1.78/8.97 | 1.34/7.88 | 2.08/ 5.06 |
| Average value | | 0.97/12.94 | 1.08/14.63 | 1.67/18.16 | 1.37/9.12 | 3.17/ 3.29 |

As can be seen from the Tab. 3, although the proposed method takes the most time in the first stage, it takes the least time by far in the second stage, with the result that the total running time of the proposed method is minimal. In the proposed method, the pixels are clustered into a grid in a bilateral space, which greatly reduces the selection of sample points, the computational complexity and the memory loss. Since the establishment of a bilateral grid is required in the first stage, our method requires more time in the first stage, but much less time in the second stage. Thus, our method requires only a small amount of user interaction to complete the color editing propagation task in a short time. The time efficiency of our method is thus highest than that of existing color editing methods. Li et al.'s [Li, Ju and Hu (2010); Chen, Chen and Zhao (2012)] method involve performing similarity judgment and color editing propagation on a single pixel; therefore, these methods take a long time. Xu et al.'s [Xu, Yan and Jia (2013)] method needs to calculate the spatial position feature, meaning that it takes an even longer time. Finally, Chen et al.'s [Chen, Li, Chen et al. (2016)] method uses a sparse dictionary for editing propagation; sparse dictionaries can reduce computations and significantly reduce time loss.

5 Conclusion

In this paper, we propose a fast image color processing method based on bilateral grid. By providing only a small amount of user interactions, different regions of the image can be edited simultaneously by combining the color features and coordinate features of the image. Using bilateral grids to cluster pixels greatly reduces the computational effort and memory loss, with the result that the time efficiency of the proposed method is improved. A grid with pixels similar to the color of the sample pixels and without marked pixels is defined as an edited grid, which can be used to accurately re-color the vertices of the edited grid. Accordingly, the proposed method realizes fast, accurate and high-quality color editing for images.

Acknowledgments: This project is supported by National Natural Science Foundation of China (No. U1836208, No. 61402053 and No. 61202439), Natural Science Foundation of Hunan Province of China (No. 2019JJ50666 and No. 2019JJ50655) and partly supported by Open Fund of Hunan Key Laboratory of Smart Roadway and Cooperative Vehicle-Infrastructure Systems (Changsha University of Science & Technology) (No. KFJ180701).

References

- Anat, L.; Dani, L.; Yair, W.** (2004): Colorization using optimization. *ACM Transactions on Graphics*, vol. 23, no. 3, pp. 689-694.
- An, X. B.; Pellacini, F.** (2008): AppProp: all-pairs appearance-space edit propagation. *ACM Transactions on Graphics*, vol. 27, no. 3, pp. 1-9.
- Amiri-Simkooei, A. R.; Mortazavi, S.; Asgari, J.** (2015): Weighted total least squares applied to mixed observation model. *Empire Survey Review*, vol. 48, no. 349, pp. 278-286.
- Bie, X. H.; Huang, H. D.; Wang, W. C.** (2011): Real time edit propagation by efficient sampling. *Computer Graphics Forum*, vol. 30, no. 7, pp. 2041-2048.
- Chen, X.; Chen, D.; Zhao, Q.** (2012): Manifold preserving edit propagation. *ACM Transactions on Graphics*, vol. 31, no. 6, pp. 1-7.
- Chen, X.; Li, J.; Chen, D.; Zhao, Q. P.** (2016): Learn sparse dictionaries for edit propagation. *IEEE Transactions on Image Processing a Publication of the IEEE Signal Processing Society*, vol. 25, no. 4, pp. 1688-1698.
- Chen, J.; Paris, S.; Durand, F.** (2007): Real-time edge-aware image editing with the bilateral grid. *ACM Transactions on Graphics*, vol. 26, no. 3, pp. 103.
- Cheng, Z.; Yang, Q.; Sheng, B.** (2015): Deep colorization. *Proceedings of the IEEE International Conference on Computer Vision*, pp. 415-423.
- Faridul, H. S.; Pouli, T.; Chamaret, C.; Stauder, J.; Reinhard, E. et al.** (2016): Colour mapping: a review of recent methods, extensions and applications. *Computer Graphics Forum*, vol. 35, no. 1, pp. 59-88.
- He, H. J.; Gui, Y.; Li, F.** (2017): Color editing of complex natural images with repeated scene elements. *Journal of Computer Science and Technology*, vol. 11, no. 11, pp. 1792-1803.
- Huang, Y. D.; Chen, C. T.** (2002): A derivation of the normal equation in fir wiener

filters. *IEEE Transactions on Acoustics Speech & Signal Processing*, vol. 37, no. 5, pp. 759-760.

Horiuchi, T.; Hirano, S. (2003): Colorization algorithm for grayscale image by propagating seed pixels. *International Conference on Image Processing*, vol. 1, pp. 454-457.

Horiuchi, T.; Kotera, H. (2006): Colorization algorithm for monochrome video by sowing color seeds. *Journal of Imaging & Technology*, vol. 50, no. 3, pp. 243.

Huang, H. Z.; Zhang, S. H.; Martin, R. R. (2015): Learning natural colors for image re-coloring. *Computer Graphics Forum*, vol. 33, no. 7, pp. 299-308.

Irony, R.; Cohen-Or, D.; Lischinski, D. (2005). Colorization by example. *Eurographics Symposium on Rendering Techniques*.

Iizuka, S.; Simo-Serra, E.; Ishikawa, H. (2016): Let there be color: joint end-to-end learning of global and local image priors for automatic image colorization with simultaneous classification. *ACM Transactions on Graphics*, vol. 35, no. 4, pp. 1-11.

Li, H.; Qin, J. H.; Xiang, X. Y.; Pan, L. L.; Ma, W.T. et al. (2018) An efficient image matching algorithm based on adaptive threshold and RANSAC. *IEEE Journals & Magazines*, vol. 6, pp. 66963-66971.

Li, Y.; Ju, T.; Hu, S. M. (2010): Instant propagation of sparse edits on images and videos. *Computer Graphics Forum*, vol. 29, no. 7, pp. 2049-2054.

Liu, S.; Song, Z. (2018): Multi-source image appearance transfer based on edit propagation. *Journal of Zhengzhou University*, vol. 39, no. 5, pp. 22-27.

Pitié, F.; Kokaram, A. C.; Dahyot, R. (2007): Automated colour grading using colour distribution transfer. *Computer Vision and Image Understanding*, vol. 107, no. 1-2, pp. 123-137.

Pellacini, F.; Lawrence, J. (2007): AppWand: editing measured materials using appearance-driven optimization. *ACM Transactions on Graphics*, vol. 26, no. 3, pp. 54.

Reinhard, E.; Adhikhmin, M.; Gooch, B.; Shirley, P. (2002): Color transfer between images. *IEEE Computer Graphics & Applications*, vol. 21, no. 5, pp. 34-41.

Shen, X. B.; Shen, F. M.; Liu, L.; Yuan, Y. H.; Liu, W. W. et al. (2018): Multiview discrete hashing for scalable multimedia search. *ACM Transactions on Intelligent Systems and Technology*, vol. 9, no. 5, pp. 1-21.

Shen, X. B.; Shen, F. M.; Sun, Q. S.; Yang, Y.; Yuan, Y. H. et al. (2017): Semi-paired discrete hashing: learning latent hash codes for semi-paired cross-view retrieval. *IEEE Transactions on Cybernetics*. vol. 47, no. 12, pp. 4275-4288.

Tai, Y. W.; Jia J. Y.; Tang, C. K. (2005): Local color transfer via probabilistic segmentation by expectation-maximization. *IEEE Computer Society Conference on Computer Vision & Pattern Recognition*, vol. 1, pp. 747-754.

Welsh, T.; Ashikhmin, M.; Mueller, K. (2002): Transferring color to greyscale images. *ACM Transactions on Graphics*, vol. 21, no. 3, pp. 277-280.

Wang, P. B. (2017): The application of radial basis function (RBF) neural network for mechanical fault diagnosis of Gearbox. *Materials Science & Engineering Conference*

Series, vol. 269, no. 1, pp. 12-56.

Wang, Z.; Bovik, A. C. (2002): A universal image quality index. *IEEE Signal Processing Letters*, vol. 9, no. 3, pp. 81-84.

Wang, Z.; Bovik, A. C.; Sheikh, H. R.; Simoncelli, E. P. (2004): Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600-612.

Xiao, B.; Wang, Z.; Liu, Q. (2018): SMK-means: an improved mini batch k -means algorithm based on mapreduce with big data. *Computers, Materials & Continua*, vol. 56, no. 3, pp. 365-379.

Xiao, Y.; Wan, L.; Leung, C. S.; Lai, Y.; Wong, T. (2015). Optimization-based gradient mesh colour transfer. *Computer Graphics Forum*, vol. 34, no. 6, pp. 123-134.

Xiang, L. Y.; Shen, X. B.; Qin, J. H.; Hao, W. (2018); Discrete multi-graph hashing for large-scale visual search. *Neural Processing Letters*.

Xiang, L. Y.; Zhao, G. H.; Li, Q.; Hao, W.; Li, F. (2018): TUMK-ELM: a fast unsupervised heterogeneous data learning approach. *IEEE Access*, vol. 6, pp. 35305-35315.

Xie, B.; Qin, J. H.; Xiang, X. Y.; Li, H.; Pan, L. L. (2018): An image retrieval algorithm based on GIST and SIFT features. *International Journal of Network Security*, vol. 20, no. 4, pp. 609-616

Xu, L.; Yan, Q.; Jia, J. (2013). A sparse control model for image and video editing. *ACM Transactions on Graphics*, vol. 32, no. 6, pp. 1-10.

Xu, K.; Li, Y.; Ju, T.; Hu, S. M.; Liu, T. Q. (2009): Efficient affinity-based edit propagation using k -d tree. *ACM Transactions on Graphics*, vol. 28, no. 5, pp. 1-6.

Zhang, D. Y.; Yin, T.; Yang, G. B.; Xia, M.; Li, L. D. et al. (2017): Detecting image seam carving with low scaling ratio using multi-scale spatial and spectral entropies. *Journal of Visual Communication and Image Representation*, vol. 48, pp. 281-291.