

## Local Color Editing Using Color Classification and Boundary Inpainting

Zhuo Su<sup>1,2\*</sup>, Xue Yang<sup>1,2</sup>, Xiaonan Luo<sup>1,2†</sup>, Dong Wang<sup>3</sup>

<sup>1</sup>National Engineering Research Center of Digital Life,  
State-Province Joint Laboratory of Digital Home Interactive Applications,  
School of Information Science & Technology, Sun Yat-sen University

<sup>2</sup>Research Institute of Sun Yat-sen University in Shenzhen

<sup>3</sup>College of Informatics, South China Agricultural University

\*suzhuoi@gmail.com, †lnslxn@mail.sysu.edu.cn

### Abstract

*Color Editing plays an important role in image processing, which aims to change the original image color according to specific image color characteristics. We present a new interactive local color editing method. The user first draws strokes specifying the target region needed to transfer color, which can be segmented using K-means clustering. Then patch-based inpainting technology is applied to achieve natural transition along the boundary. In the course of color editing, only the color of the pixels in this segmented region is transferred, while others including the boundary region and the background remain unchanged. The experimental results show that our method can achieve a visually satisfying local color editing results.*

### 1. Introduction

Color transfer is an important image editing technology, widely used in movie production, photo processing, art design, etc. The interactive color transfer technique allows to change the original image color style in terms of user's intent, which is more flexible than the automatic color transfer. In this paper, we focus on local color editing. On the basis of the interactive technology, the region of interest (ROI) is segmented out through color classification, and then we apply the color transfer in the ROI. However, inaccurate segmentation caused by insufficient sampling, would result in unnatural effects along the boundary. In this paper, we propose a local color editing method transferring colors from reference image to the ROI of the source image, exploiting color classification for segmentation and applying boundary correction for natural transition between the ROI and the background.

The interactive local color editing presented in this paper has three stages: user-specified target region seg-

mentation, region boundary correction and color transfer. Users draw some strokes on the source image and the reference image to specify the ROI needed to transfer color and the reference color respectively. The pixels of the source image are classified using the K-means based color classification with user specifies, which can roughly cut the ROI out. Then the boundary is modified by inpainting. In the phase of color transfer, only the ROI pixels are applied color transfer. With boundary correction, the transition from the ROI to the background is much more natural and believable.

The mainly contributions of this paper include these following three aspects: 1) Proposing the color classification for region segmentation. 2) Introducing inpainting into boundary correction. 3) The effectiveness and applicability of the proposed method are proved by our experimental results.

### 2. Related work

Generally speaking, color transfer can be divided into global and local color transfer. Reinhard et al. [5] proposed a global color transfer method imposing a reference image's color characteristics into another source image using statistical analysis. The color of the source image is transferred according to the mean and standard deviation of the color values in the source and reference image, with keeping the image content naturally. A variety of methods are proposed to improve Reinhard's color mapping method [5] to achieve better color transfer results, e.g.[2][6][7][8].

There have been proposed many algorithms for local color transfer. An important color transfer method is to propagate the new color of pixels in the strokes to other pixels of the source image [4][7]. Li et al. [4] casted the editing interpolation to the radial basis function interpolation, improving the efficiency of color transfer for large image and video. Wang et al. [7]

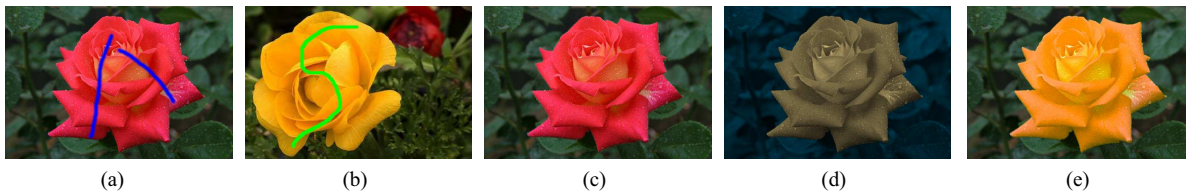


Figure 1: Overview of our method. (a) Source image with marked strokes. (b) Reference image with the corresponding stroke. (c) Result of boundary correction using inpainting. (d) Result of region extraction in pseudo colorization. (e) Final result.

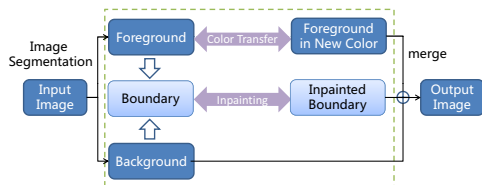


Figure 2: The pipeline of our local color editing.

presented the mean value coordinates based local color transfer method, applying Reinhard’s linear color transfer method [5] and the discrete mean value coordinates model.

Image segmentation is a simple and feasible solution for local color transfer [6][8]. Wen et al. [8] proposed a system for image enhancement by using interactive color transfer algorithm. Firstly, an improved graph cut algorithm was presented for segmentation. Then a Gaussian probability-weighted color transfer algorithm was applied between corresponding image region pairs. At the end, the authors used a gradient based smoothness term to improve the results, which improves the controllability of the method.

However, currently there is no perfect segmentation algorithm which is always accurate. Visual unsatisfied or even artifacts arise along the boundary between the ROI and the background, due to the contrast variance of the new color and the source color. To overcome this problem of color transfer, we introduce inpainting into local color editing, which greatly improves the visual effect along the boundary and achieves natural color transfer results.

### 3. User stroke based local color editing

The local color editing method we proposed exploits color classification, image segmentation, boundary correction, and color transfer. The pipeline of our method is shown in Fig.2. As previous mention, user should draw several strokes both on the source image and the reference image to specify the color desired to be transferred and the new color. At the second stage, we apply the K-means based color classification method to segment the ROI out, and the boundary of which is cor-

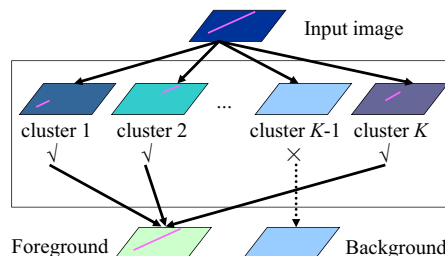


Figure 3: Region extraction via color classification. The ROI and the amount of color are specified by user strokes and assignment.

rected by inpainting in the following stage. Finally, a color transfer operation is applied to transfer colors from the reference image to the source image.

#### 3.1. User stroke

What are necessary for color transfer in the last phase are the original color from the source image and the new color from the reference image. It is easy to get these colors by drawing strokes on images. Specially, those strokes on the source image ought to involve all the colors that need to transfer in the ROI, to ensure the accuracy of color classification.

The pixels of source image can be denote as  $\langle p, v, C, R \rangle$ ,  $p$  presents the coordinates of the pixel,  $v$  is the color vector,  $C$  is the color category, and  $R$  labels which region the pixel belongs to: 0 (background), 1 (foreground), or -1 (boundary inpainting region). Given a source image  $I_s$  and a reference image  $I_r$ , we denote the final color transfer result as  $I_t$ .  $P_s = \{(p_i, v_i, C_i, 1) | i = 1, \dots, n_1\}$  and  $Q_r = \{(p_j, v_j) | j = 1, \dots, m\}$  respectively present the set of pixels under the user strokes in the source image and the reference image. Generally,  $P_s \in I_s$ .

#### 3.2. Region extraction by color classification

It is necessary to segment the ROI out for local color editing at first. We first use K-means clustering algorithm to divide source image into different regions based on the color categories. According to the original colors from user strokes, we merge corresponding re-

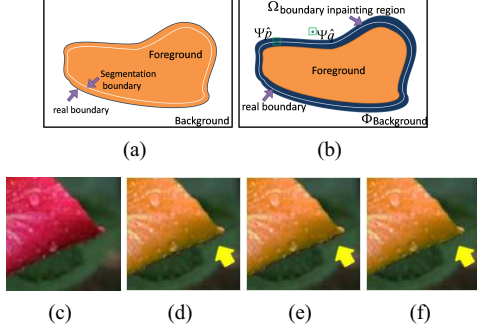


Figure 4: Boundary correction using inpainting. (a) Result of image segmentation. (b) Schematic illustration for boundary correction discussed in section 3.3. (c) Source image. (d) Result using K-means classification based segmentation only. (e) Result using image matting [3] mended segmentation. (f) Result using inpainting [1].

gions to get a rough ROI. The procedure of this region extraction method is shown in Fig.3.

We first convert the image to Lab color space, and apply K-means clustering algorithm to classify all the pixels. Select  $K$  centroids, denote as  $z_1(1), z_2(1), \dots, z_K(1)$ . In every iteration, we calculate the distance  $dis(x, z_k(1))$  between pixel  $x$  and the centroid  $z_k(1)$ ,  $k = 1, 2, \dots, K$ . Here, the distance  $dis(i, j)$  is defined as follows

$$dis(i, j) = \sqrt{(l_i - l_j)^2 + (\alpha_i - \alpha_j)^2 + (\beta_i - \beta_j)^2}. \quad (1)$$

At the end of classification, every pixel's color category is determined. If pixel  $i$  and pixel  $j$  belong to the same category,  $j \in P_s$ ,  $R_i = 1$ , otherwise  $R_i = 0$ . Through this process, we can segment out a rough target region which has the same color with the user strokes.

### 3.3. Boundary Correction

Color classification can not get satisfied segmentation result, when the original image has various colors. These results generally arouse residuals along the edge, which has significant influence on color transfer. Image matting is widely used in object extraction, elaborately separating the foreground and background of the image. However, in the case that obscure boundary appearing between the ROI and the background, image matting is incapacitated for edge correction neither.

Image inpainting is an important research field in image restoration, removing objects and replacing them with visually plausible backgrounds. By replaced by textures sampled from the background of the image, the restored target region visually satisfies the human vision to a certain degree[1]. Here, we introduce inpainting

into our local color editing to achieve visually satisfying results near obscure boundary.

As shown in Fig. 4(b), we expanded ROI 5-10 pixels to get a narrow loop, defined as boundary inpainting region  $\Omega = \{(p_l, v_l, C_l, -1) | l = 1, \dots, n_2\}$ , and replace  $\Psi\hat{p}$ , a patch centered on pixel  $p$ ,  $p \in \Omega$ , with the most similar patch from the background  $\Phi$  to propagate image texture. The target patch  $\Psi\hat{q}$  is defined as [1]

$$\Psi\hat{q} = \arg \min_{\Psi q \in \Phi} d(\Psi\hat{p}, \Psi q), \quad (2)$$

where patch  $\Psi\hat{p}$  has highest priority on the fill front, and the distance  $d(\Psi\hat{p}, \Psi q)$  between  $\Psi\hat{p}$  and  $\Psi q$  is defined as the sum of squared differences of those already filled pixels in the two patches.

Then each pixel in  $\Omega$  is updated to  $\Omega = \{(p_l, v'_l, C_l, -1) | l = 1, \dots, n_2\}$ .

### 3.4. Color transfer operator

We apply the method proposed by Reinhard et al. [5] to transfer colors. Define the mean and variance of the ROI  $I_s'$  and the set of pixels of the reference image stroke  $Q_r$  as follows

$$\mu_s = \frac{1}{n'} \sum_{i=1}^{n'} v_i, \sigma_s = \frac{1}{n'} \sum_{i=1}^{n'} (v_i - \mu_s)^2. \quad (3)$$

$$\mu_r = \frac{1}{m} \sum_{j=1}^m v_j, \sigma_r = \frac{1}{m} \sum_{j=1}^m (v_j - \mu_r)^2. \quad (4)$$

According to the formula provided in the Reinhard's method [5], we can calculate the new color of pixels  $I_s' = \{(p_i, \bar{v}_i, C_i, 1) | i = 1, \dots, n'\}$  by

$$\bar{v}_i = \frac{\sigma_r}{\sigma_s} (v_i - \mu_s) + \mu_r. \quad (5)$$

Consequently, we get the final result  $I_t = \{(p_i, \bar{v}_i, C_i, 1) | i = 1, \dots, n'\} \cup \{(p_l, v'_l, C_l, -1) | l = 1, \dots, n_2\} \cup \{(p_j, v_j, C_j, 0) | j = 1, \dots, N - n' - n_2\}$ .

## 4. Experimental results and discussion

In this section we demonstrate our experimental results and the comparisons with previous methods. The experimental input is a source image and a reference image with the user strokes on corresponding image region. Note that it is unnecessary to mark the unchanged region of source image, which on contrary is essential for previous methods. Our method can be extended to the case of multiple strokes or multiple reference images [8].

Most of the experiments with parameters  $K = 10$  and the window in boundary inpainting set to be  $3 \times 3$  achieve good results. For K-means based color classification for segmentation,  $K$  must enable accurate color classification. Smaller setting of  $K$  is for simple color

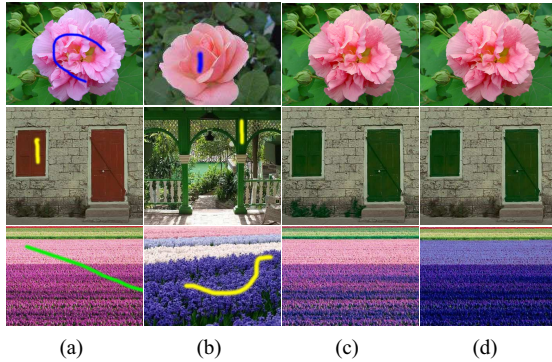


Figure 5: Comparison with Wang’s method in [7]. (a) Source image. (b) Reference image. (c) Result of Wang’s method. (d) Result of our method with  $K = 10$ .

distribution. The boundary inpainting region depends on the difference of segmentation boundary and the real target boundary, which must cover the real target boundary. In the stage of boundary inpainting, small size of patch window allows better propagation of the background’s structure and color, which is time-consuming however.

We first compare the results of Wang’s method [7]. It’s hard to tell the difference between the results as shown in the first row of Fig.5(c) and (d). However, as shown in the rest two rows of Fig.5(c) and (d), the background is changed partially and the color in source image stroke can still be seen. Ours transferring corresponding color entirely to the whole ROI, is more natural and satisfying on the contrary.

Fig.6 shows the two comparison of our method and the edit propagation method proposed by Li et al. [4]. In the first experiment, two kinds of strokes are given: one specifies the color of pixels to be transferred, one specifies those remain unchanged which is not demonstrated in Fig.6(a). The background of the result in Fig.6(c) has been changed greatly, on the contrary, as shown in Fig.6(d), our result’s background is well preserved. Whereas, in second experiment, only one kind of strokes is given specifying the color of those pixels will be changed into the new color. The result of second experiment in Fig.6(c) is far from user’s expectation, while ours is much better.

## 5. Conclusion

In this paper, we propose a new local color editing method exploiting color classification and linear color transfer. We particularly apply inpainting for natural transition between the target region of user’s interest and the background. Our experiments show that our method can segment the target region out to transfer color, according to the users’ expectations, and achieve

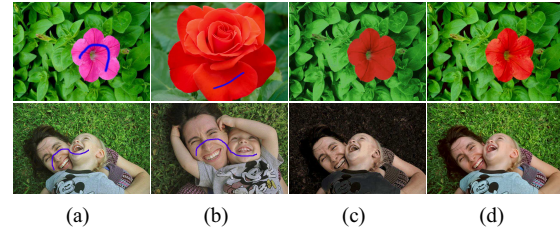


Figure 6: Comparison with Li’s method in [4]. (a) Source image. (b) Reference image. (c) Result of Li’s method with  $\sigma_c = \sqrt{0.2}$ . (d) Result of our method with  $K = 5$ .

natural results. Finally, Our method need to make a further improvement on the discrimination of similar colors.

**Acknowledgment:** This research is supported by NSFC-Guangdong Joint Fund (U0935004, U1135003), the National Key Technology R&D Program (2011BAH27B01, 2011BHA16B08), the Major Science and Technology Projects of Guangdong Province (2011A080401007), and the Industry-academy-research Project of Guangdong (2011A091000032). Thank all the authors providing images for our experiment, as well as adamnsinger.

## References

- [1] A. Criminisi, P. Perez, and K. Toyama. Object removal by exemplar-based inpainting. In *Computer Vision and Pattern Recognition, 2003.*, volume 2, pages 721–728, june 2003.
- [2] W. Dong, G. Bao, X. Zhang, and J.-C. Paul. Fast local color transfer via dominant colors mapping. In *ACM SIGGRAPH ASIA 2010 Sketches*, pages 46:1–46:2, 2010.
- [3] K. He, J. Sun, and X. Tang. Guided image filtering. In *Proceedings of the 11th European conference on Computer vision: Part I*, pages 1–14, 2010.
- [4] Y. Li, T. Ju, and S.-M. Hu. Instant propagation of sparse edits on images and videos. *Computer Graphics Forum*, 29(7):2049–2054, 2010.
- [5] E. Reinhard, M. Ashikhmin, B. Gooch, and P. Shirley. Color transfer between images. *Computer Graphics and Applications, IEEE*, 21:34–41, September 2001.
- [6] Y.-W. Tai, J. Jia, and C.-K. Tang. Local color transfer via probabilistic segmentation by expectation-maximization. In *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '05)*, pages 747–754, 2005.
- [7] D. Wang, W. Jia, and G. Li. Color transfer using scattered point interpolation. In *Proceedings of the 10th International Conference on Virtual Reality Continuum and Its Applications in Industry*, pages 163–170, 2011.
- [8] C.-L. Wen, C.-H. Hsieh, B.-Y. Chen, and M. Ouhyoung. Example-based multiple local color transfer by strokes. *Computer Graphics Forum*, 27(7):1765–1772, 2008.