Proceedings of the Fifth IIEEJ International Workshop
on Image Electronics and Visual Computing 2017
Da Nang, Vietnam, February 28- March 3, 2017

Lazy Colorization: A Scribble-based Tool for Color Elements Selection

Shun Toukairin Yoshinori Dobashi Tsuyoshi Yamamoto
Hokkaido University

ABSTRACT

In this paper, we propose a new paint tool, *Lazy Colorization,* which is a scribble-based tool for recoloring desired color elements among the scribble-covered elements. *Lazy Colorization* helps a user to paint multiple elements. Our algorithm has two steps. First, by considering shape, size, color, etc. of elements, our algorithm selects the candidates that the user seems to select from elements covered by the user-drawn scribble. Second, our algorithm ranks them on the basis of the user's scribble, and then identifies the top-ranked candidate as the user's most desired elements(s). As our tool takes account of color information, it is more effective in the case of redrawing color elements.

1. INTRODUCTION

When we use a paint tool, we will often use a so-called paint bucket tool to paint an element (i.e., a closed region). This tool fills a region with similar colored pixels with a specified color. However, this tool is not applicable to paint multiple elements at a time. Thus, a new element selection tool that helps people to paint multiple elements more easily at once has been developed.

Lazy Selection [2] is a scribble-based tool for quick selection of one or more desired shape elements by roughly stroking through the elements. By grouping elements with Gestalt rules, and then ranking them with respect to the scribble, this tool fills multiple elements that the user wants to paint. However, this method has a problem; it supports only a line drawing.

In this paper, we address the above problem by introducing color information into the group error used for grouping elements. We also propose a method to edit the color of a color image using our colored element selection tool.

2. OVERVIEW OF LAZY SELECTION ALGORITHM

In this section, we briefly explain *Lazy Selection* [2], which our method is based on.

Lazy Selection [2] is a scribble-based tool for smart shape elements selection. First, this method calculates the grouping error f_{ij} between a pair of elements i and j, and decides the candidates of the elements to be painted. The grouping error f_{ij} reflects similarity, proximity, common region principle derived from Gestalt's psychology, which describes how the human brain recognizes objects as a whole. In [2], the grouping error is formulated as follows:

$$f_{ij} = \left(1 - e^{-\frac{|\lambda_i^1-\lambda_j^1|}{|\lambda_i^1+\lambda_j^1|}-\frac{|\lambda_i^2-\lambda_j^2|}{|\lambda_i^2+\lambda_j^2|}-\frac{|c_i-c_j|}{|c_i+c_j|}}\right)(1-\gamma e^{-g_{ij}})b_{ij}, \quad (1)$$

where λ_i^k is the $k-\text{th}$ eigenvalue obtained by applying PCA to element i. c_i is defined as the ratio of the squared perimeter to the area of element i. g_{ij} is the Euclidean distance between the centers of elements i and j, normalized by the average of all pairwise distances. γ is a weight to adjust the contribution of proximity principle. b_{ij} controls the ease of grouping; the smaller b_{ij} is, the easier the two elements are grouped. We use $b_{ij} = 0.7$ if elements are inside a common region; $b_{ij} = 1$, otherwise.

Next, a tree structure is constructed by grouping the elements recursively. Each element is firstly assigned as a leaf node in the tree. Then, grouping errors between nodes are computed and nodes with the minimum grouping error are grouped to create a parent node of them. The grouping error between nodes is defined as the maximum grouping error among all pairs of elements assigned to the nodes. Each node in the tree corresponds to a candidate of element(s) that the user seems to select. Let C denote a set of candidates.

Second, the system matches the candidates $\mathrm{c} \in \mathrm{C}$ and user's scribble, and choose the candidate $\mathrm{c} \in \mathrm{C}$ ranked highest as the user's intention. Let P_s and P_c denote two sets of points sampled for the scribble and the candidate $\mathrm{c} \in \mathrm{C}$, respectively. Using them, the matching error function $r(c)$ is defined by the following equation:

$$r(c) = \beta(\hat{\mu}_{s\to c} + \hat{\mu}_{c\to s}) + (1-\beta)(\hat{\sigma}_{s\to c} + \hat{\sigma}_{c\to s}), \quad (2)$$

where $\hat{\mu}_{i\to j}$ and $\hat{\sigma}_{i\to j}$ are the mean and the standard deviation of the closest distance from every point in P_i to P_j, normalized by their average among all the candidates, respectively. β is a variable with respect to the speed of scribbling. When $r(c)$ is smaller, the scribble and the candidate $\mathrm{c} \in \mathrm{C}$ are matched.

Since a scribble should pass through the elements that the user wants to paint, we assume that the candidate $\mathrm{c} \in \mathrm{C}$ matched with the scribble is what the user desires. Therefore, the top-ranked candidate $\mathrm{c} \in \mathrm{C}$, that is, the candidate $\mathrm{c} \in \mathrm{C}$ having the smallest $r(c)$ is chosen as the user's desire.

3. PROPOSED METHOD

Lazy Selection [2] enables us to paint multiple elements at once. However, we can't obtain the desired result for colored elements. For example, if a user provides a scribble shown in Figure 1(b) in order to change the red part of the polka dot pattern (see Figure 1(a)) to green, all the polka dots turn into green as shown in Figure 1(c). We propose a method to address this problem by adding color information to the grouping condition.

In the proposed method, a color histogram is used in order to compute the similarity between a pair of color elements. Using Bhattacharyya coefficient for calculating the similarity between two histograms, the following new grouping error F_{ij} is defined.

$$F_{ij} = f_{ij} + \left(1 - e^{(H_{ij}-1)}\right), \quad (3)$$

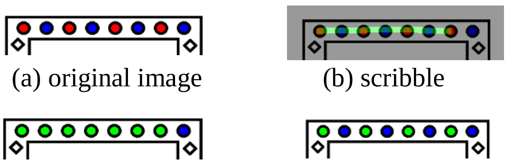
where H_{ij} is the Bhattacharyya coefficient between the color histograms of the elements i and j. We compute the histograms by quantizing the colors in the element into 64 colors. A new grouping error F_{ij} becomes larger when the color histograms between the elements i and j are different, making it difficult to group them.

Next, we describe a method for editing color images by using the proposed method. We assume that the segmentation of the input image has been finished. That is, the image is represented as a collection of elements (closed regions) with similar colors. The color histogram for each element is pre-calculated and used for computing the grouping error expressed by Eq. (3). In order to modify the colors of the elements selected by the user, we use HSV color space. The user can change the hue component of the color.

4. RESULT

We used a computer with CPU: Core(TM) i7-4770K 3.50KHz (Memory: 8.00GB). Our method runs in real-time, making it possible to interactively edit color images.

Figure 1 shows the result obtained by the proposed method using the new grouping error. Even if the user selects all the red and blue polka dot pattern as shown in Figure 1(b), only the red part is painted as in Figure 1(d). This selection is not included even in the candidates in the conventional method. Our method successfully groups the elements reflecting color information.

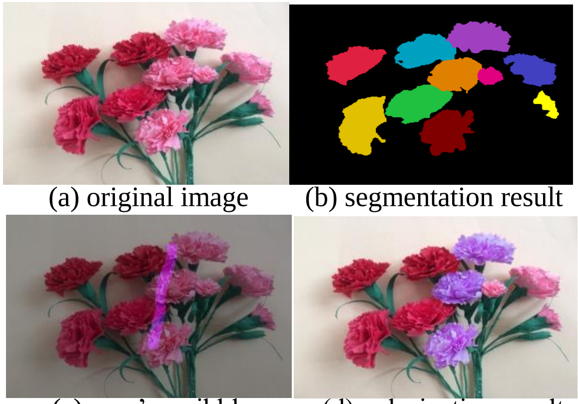


(a) original image (b) scribble

(c) Lazy Selection (d) Proposed method

Figure 1. An example of selecting colored elements.

Figure 2 shows the result of applying the proposed method to the color image. The segmentation of the image has been done in a preprocess as shown in Figure 2(b). In this example, the user provided a scribble to pass through five red and pink flowers as shown in Figure 2(c) and, the color of pink flowers was modified as shown in Figure 2(d). This example demonstrates that our method is successfully applied to color pictures.



(a) original image (b) segmentation result

(c) user's scribble (d) colorization result

Figure 2. Result applied to color picture.

5. CONCLUTION

We have developed a method for selecting color elements reflecting user's intention by adding color information to the existing method, *Lazy Selection* [2]. Moreover, we have also presented a method to apply our method to color pictures by using element detection image. We believe that our method makes image editing easier.

There are some issues that should be addressed in the future. First, in some cases, the candidate with the smallest matching error function may not be the result desired by the user. However, this problem could be easily resolved by adding color information to the matching error function. Next, in addition to the color information, we think that our method cloud be improved by taking into account the Gestalt's continuity law. Since the image segmentation sometimes fails and the user cannot select desired elements, we would like to develop a method that allows the user to select objects without the image segmentation. Finally, since our method can't be applied to grayscale images, we would like to consider this in further research.

6. REFERENCES

[1] KUBOVY, M., AND VAN DEN BERG, M. 2008. The whole is equal to the sum of its parts: a probabilistic model of grouping by proximity and similarity in regular patterns. Psychological review 115, 1, 131.

[2] Pengfei Xu , Hongbo Fu , Oscar Kin-Chung Au , Chiew-Lan Tai, Lazy selection: a scribble-based tool for smart shape elements selection, ACM Transactions on Graphics (TOG), v.31 n.6, November 2012