

# Overview of Scribbled-Based Colorization

Simeng Li<sup>1\*</sup>, Qiaofeng Liu<sup>2\*</sup>, Haoyu Yuan<sup>3\*</sup>

<sup>1</sup>Beihang University, Beijing, China

<sup>2</sup>Southeast University, Nanjing, China

<sup>3</sup>Shenzhen University, Shenzhen, China

Email: 1907096379@qq.com

**How to cite this paper:** Li, S. M., Liu, Q. F., & Yuan, H. Y. (2018). Overview of Scribbled-Based Colorization. *Art and Design Review*, 6, 169-184.

<https://doi.org/10.4236/adr.2018.64017>

**Received:** September 3, 2018

**Accepted:** October 20, 2018

**Published:** October 23, 2018

Copyright © 2018 by authors and Scientific Research Publishing Inc.

This work is licensed under the Creative Commons Attribution-NonCommercial International License (CC BY-NC 4.0).

<http://creativecommons.org/licenses/by-nc/4.0/>



Open Access

---

## Abstract

Hand-coloring is very useful when people draw pictures or produce movies, but it is also very time-consuming and expensive. Therefore, with the development of segmentation algorithms and deep learning, a lot of computer-assisted colorization algorithms have been invented, including user-guided colorization, semi-automatic colorization and automatic colorization. In this survey paper, we will only focus on scribble-based colorization in user-guided colorization and review the development on it. We will discuss various principles and methods among different kinds of scribble-based colorization algorithms. We will also compare visual result and performance of algorithms among colorization on grayscale image, colorization on sketch and colorization with networks to conclude their advantages and disadvantages.

## Keywords

Scribble-Based Colorization, Energy Function, Color Blending, Convolutional Neural Network

---

## 1. Introduction

**Colorization** refers to a process of adding colors to black-and-white images (with grayscale information), sketches or even monochrome motion pictures. The traditional hand-coloring methods are very time-consuming and require the effort of a whole group of professional artists. Therefore, many computer-assisted colorization algorithms have been invented to make the process of colorization more efficient.

Computer-assisted colorization can be implemented in a large range of areas. For example, since the colorization can only be finished by hand, colorizing a black-and-white movie cost more than \$3000 per minute of running time to co-

\*they are all co-first authors, signature by the alphabetical order of last names.

lorize a movie in 1987 according to a report in *Popular Mechanics*. To reduce the cost, computer-assisted colorization algorithms are invented, offering film-makers a less expensive way of producing technicolor and bringing more realistic visual effects to the audience.

As a second example, it takes a lot of efforts to colorize the sketch picture after a painter finishes the creative sketch because painters need to find the proper color first, which is very difficult for the novice painters, and then colorize every region in the picture manually, which is very time-consuming. To improve the visual effect and the efficiency of drawing, many automatic colorization algorithms are desirable by which the color of the sketch picture can be assigned automatically and changed quickly according to the painters' preference.

According to the participation level of users, computer-assisted colorization methods can be divided into three domains: automatic, semi-automatic and user-guided colorization methods. Automatic colorization methods are recent approaches by which the monochrome pictures are directly colorized by training a Convolutional Neural Networks (CNN) with a large-scale image collection. Semi-automatic colorization methods denote approaches by which color pattern is transferred from one or more reference images to the input monochrome picture. User-guided colorization methods are approaches by which users can directly decide color of the corresponding region.

In this paper, we will focus on scribble-based colorization in user-guided colorization. First, we will introduce the history of the scribble-based colorization briefly. Then, we will elaborately talk about three areas of scribble-based colorization, including grayscale image colorization, sketch colorization and colorization with networks. In each area, we will make analyses of differences of principles, methods, performance among the colorization algorithms. Finally, we will draw a conclusion about their advantages and disadvantages. We hope that these analyses can become a useful resource for the scribble-based colorization and computer graphic community.

## 2. A Brief History of Scribble-Based Colorization

In the earlier days, the process of colorization was divided into two separated parts: segmentation and filling. However, the error ratio in segmentation remains to be high, which means that a lot of user-interventions are needed to fix the errors, making colorization a tedious, time-consuming and expensive task. To reduce user-interventions, Anat Levin and her colleagues designed a method that focused on assigning colors of pixels utilizing the optimization algorithm according to the similarities of intensities, improving the accuracy in colorization (Levin et al., 2004). Nonetheless, the processing time is still too long, and users still need to be very careful when choosing the colors and strokes' positions. The non-iterative method combined with the adaptive edge extraction (Huang et al., 2005) was proposed to reduce the running time of colorization optimization and color bleeding effects. Different with previous methods, Yatziv & Sapiro (2006) presented the idea of color blending, which means that the

chrominance value of a pixel is the result of contribution from given colors. To achieve a better visual effect when user strokes are sparse, Luan et al. (2007) proposed to consider both the neighboring pixels with similar intensity and remote pixels with similar textures. For the same purpose, for the same purpose, Xu et al. (2013) applied the probability distribution to find out the most confident stroke color for every pixel and thus decide the color of pixel.

However, intensity-based colorization methods used in colorizing grayscale images may fail to colorize sketch because manga has no grayscale information. Qu et al. designed a method that can propagate color through pattern-continuous regions as well as intensity-continuous regions (Qu et al., 2006). Later, Sykora and colleagues pointed out the limitations in Qu et al. and formulated their ideal painting tool by converting the properties to an energy minimization framework in order to colorize black-and-white cartoons more conveniently (Sýkora et al., 2009). With the development of deep learning, Zhang et al. (2017) incorporated U-Net structure into their colorization framework to reduce user's interventions and improve the visual effect of colorization.

### 3. Colorization on Grayscale Image

#### 3.1. Colorization with Many User Interventions

##### 3.1.1. Principle

The principle for colorization that neighboring pixels with similar intensities should have similar colors was first proposed by Levin et al. (2004) and then was used by Huang et al. (2005). Yatziv & Sapiro (2006) proposed another method based on Color Blending, and simply assumed that the closer two pixels are in the intrinsic distance (geodesic distance), the more similar chrominance value they may have.

##### 3.1.2. Colorization Methods

Levin et al. and Huang et al. used YUV color space where  $Y(x, y)$  represents the luminance of the pixel whose ordinate is  $(x, y)$  and  $U(x, y)$  and  $V(x, y)$  represent the color volumes, regarding value of  $Y(x, y)$  to be the input and values of  $U(x, y)$  and  $V(x, y)$  to be the output. Output of  $U(x, y)$  and  $V(x, y)$  is decided by minimizing the cost function:

$$J(U) = \sum_p \left( U(p) - \sum_{q \in N_s(p)} w_{pq}^s U(q) \right)^2 \quad (1)$$

$$J(V) = \sum_p \left( V(p) - \sum_{q \in N_s(p)} w_{pq}^s V(q) \right)^2 \quad (2)$$

where  $N_s(p)$  is the set of 8 spatially neighboring pixels around pixel  $p$ . When the cost function  $J(U)$  and  $J(V)$  reduce to the minimum, the corresponding  $U$  and  $V$  will be the final color volumes for pixel  $p$ . It's worth mentioning that Levin et al. and Huang et al. had different definitions about the weight function  $w_{pq}^s$ . The formula proposed by Levin et al. is:

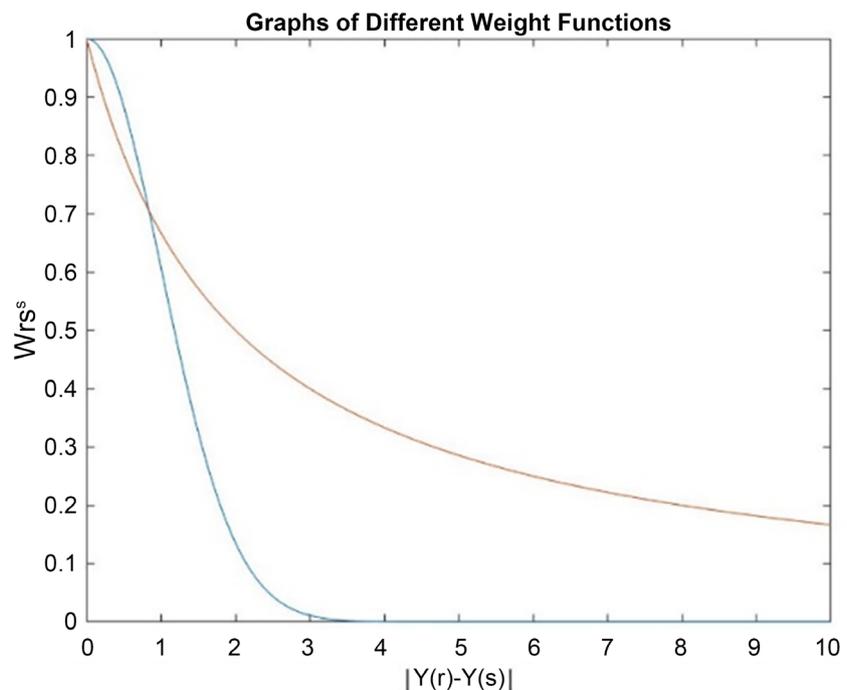
$$w_{pq}^s \propto e^{-\frac{(Y(p)-Y(q))^2}{2\sigma_p^2}} \quad (3)$$

where  $\sigma_p^2$  is the variance of luminance in the window around pixel  $p$ . However, Huang et al. designed a new function of weight:

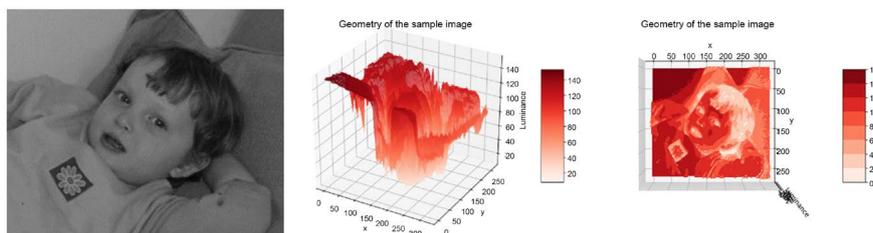
$$w_{pq}^s = \frac{1}{1 + \frac{|Y(p) - Y(q)|}{\sigma_p^2 + 1}} \quad (4)$$

By assuming that  $\sigma_p^2 = 1$ , we can get two different functions about  $w_{pq}^s$  and  $|Y(p) - Y(q)|$  and draw two function graphs according to Equation (3) and Equation (4). From the graphs, we find that when  $|Y(p) - Y(q)|$  (the luminance difference) becomes larger, the function value defined by Equation (3) converges to zero much faster than the function graph defined by Equation (4), proving that the weight defined by Equation (3) will not change when intensity difference between  $Y(p)$  and  $Y(q)$  is very large. After analyzing different function graphs of those two definitions (**Figure 1**), we can conclude that the weight function is more proportional to the color differences under Equation (4) and thus the color can be changed proportionally to the luminance differences.

Rather than following the optimization framework, Yazitv et al. proposed a new framework based on Color Blending which means the chrominance of a pixel is a weighted average of the chrominances in the user strokes. The weight of a specific chrominance is given by a carefully designed weighted function which takes the minimum distance from the pixel “to this chrominance” as an input. A very important but unstated idea beyond this paper is that a grayscale image can be seen as a two-dimensional representation of a special geographic surface. (**Figure 2**) In this setting, the geodesic distance is a common and proper



**Figure 1.** Graph of Different Weight Functions. The green line represents the Equation (3) while the red line represents the Equation (4).



**Figure 2.** This figure displays the geometry (or topography) of the sample grayscale image. From left to right: sample image and the geometry of it (a general view and a top view, respectively).

choice for measuring the distance between two pixels, but the efficiency may suffer because calculating the geodesic distance involves complex integrals (Yatziv & Sapiro, 2006). However, since the grayscale image is actually not an ideal geographic surface but a collection of discrete spatial points, it can be simplified as a nonplanar undirected graph with weights at the edges taking on the Euclidean Distance between two pixels (vertices). Therefore, a significant speed-up can be achieved by computing the minimum distance using Dijkstra's algorithm, which is sufficient for the precision. Combining it with other tricks (Yatziv & Sapiro, 2006) yields a good tradeoff between efficiency and quality.

### 3.1.3. Performance

The method proposed by Levin et al. is very time-consuming, since the optimization included hundreds of iterations. To reduce the running time of optimization, Huang et al. made an improvement on this method. There are four steps in the new method: 1) Triangulate the pixels with assigned colors using the Delaunay triangulation algorithm; 2) Calculate the colors of pixels along the edges of those triangles using the color volume of corresponding end vertices; 3) Initialize the color volumes of the pixels inside the triangles according to the color volumes of edges. 4) Run the colorization optimization algorithm using Equation (1) and Equation (2). **Figure 3** (Huang et al., 2005) shows the result using those two methods. As a result, with the non-iterative algorithm proposed by Huang et al., we can get pictures with similar PSNR. Further, by using Color Blending instead of the optimization framework, the algorithm in Yatziv & Sapiro (2006) is approximately 20 times faster than the algorithm proposed by Levin et al., while it is still able to achieve a better visual effect (**Figure 4**).

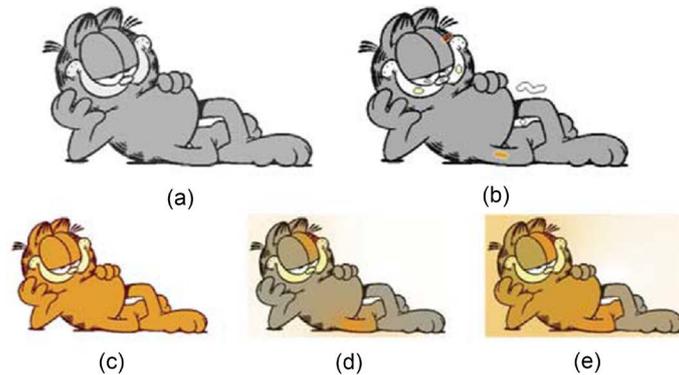
## 3.2. Colorization with Less User Interventions

### 3.2.1. Principle

When colorizing natural image, those methods need user to input many strokes and to draw them elaborately. To reduce user's efforts, Luan et al. (2007) thought it was necessary to consider not only the neighboring pixels with similar intensities but also the remote pixels with similar texture. For the same purpose, Xu et al. (2013) applied the probability distribution to find out the most confident stroke color for every pixel and thus decided the color of pixel.



**Figure 3.** Output comparisons between the iterative algorithm and non-iterative algorithm. (a) The output by iterative method ( $PSNR_U = 36.3$ ,  $PSNR_V = 37.1$ ); (b) The output by non-iterative method ( $PSNR_U = 35.7$ ,  $PSNR_V = 38.4$ ).



**Figure 4.** (a) Source Image; (b) Image with scribbles; (c) Yatziv & Sapiro (2006) (1.76s); (d) Levin et al. (2004) (a faster version) (22.8s); (e) Levin et al. (2004).

### 3.2.2. Colorization Methods

By adding that remote pixels with similar textures should be colorized with similar color, a new colorization approach (Luan et al., 2007) realized colorization with only a few strokes. After studying failure examples, Luan et al. decided to colorize pixels that are near to edges according to the texture similarity and colorize pixels in smooth regions according to the intensity similarity. Consequently, Luan designed an energy function framework:

$$E = \sum_{p \in I} (\lambda(p) E_1 + (1 - \lambda(p)) E_2) \quad (5)$$

where  $\lambda(p)$  is the weight map;  $E_1$  and  $E_2$  are textural term and spatial term:

$$E_1 = \sum_{q \in N_t(p)} w_{pq}^t \|L(\mathbf{C}; p) - L(\mathbf{C}; q)\| \quad (6)$$

$$E_2 = \sum_{q \in N_s(p)} w_{pq}^s \|L(\mathbf{C}; p) - L(\mathbf{C}; q)\| \quad (7)$$

where  $N_t(p)$  is spatially neighboring pixels of  $p$ ,  $N_s(p)$  is texturally neighboring pixels of  $p$ ;  $\mathbf{C}$  represents all the colors:  $\mathbf{C} = [C_1, C_2, \dots, C_n]^T$ ;  $L(\mathbf{C}; p)$  is the likelihood function. Besides,  $E_1$  and  $E_2$  are defined under the constrain:  $L(\mathbf{C}; p) = [0, \dots, 1_k, \dots, 0]^T$  when  $p$  belongs to the stroke with color  $C_k$ . When the pixel  $p$  is close to the edges,  $\lambda(p)$  will be larger and thus the

spatial term will be less influential than the textural term.  $N_t(p)$  and  $w_{pq}^t$  are defined in the same way as before, while  $N_s(p)$  is defined by using k-means to cluster on patches according to their similarity in appearance and  $w_{pq}^s$  is defined according to both the textural similarity and spatial continuity. By iteratively updating  $L(C; p)$  until it is fixed, we can solve the energy function and finally get the likelihood vector where the color with the maximum likelihood will be assigned to the corresponding pixel. At last, users can adjust the colors to make the image more vivid only by choosing a few pixels to assign the color. According to those assigned colors and intensity information, the color of other pixels can be produced by interpolations.

Later, Xu et al. proposed to reduce the user strokes from the perspective of designing an adaptive feature space. A feature vector of a pixel in the feature space contains the information of position, color, user control confidence and propagation result in previous iteration. According to the closeness in feature space between pixels, the weighted sum of Gaussian functions is generated by all the pixels in user strokes. After some derivations, to choose the color with maximum probability, Xu et al. presented the objective function:

$$E(p) = (p - Wg)^T (p - Wg) + \lambda_s p^T Lp \quad (8)$$

where  $g$  is the user input color vector ( $g_j$  is non-zero only when  $j \in \Omega_{\text{stroke}}$ );  $p$  is the output color vector;  $L$  is the sparse Laplacian for regularization;  $p^T Lp$  is the smoothness term (Lischinski et al., 2006);  $\lambda_s$  determines the smoothness level;  $W$  is the weight matrix relative to the feature vector.

### 3.2.3. Visual Effect

**Figure 5** shows that by inputting a lot of user strokes (**Figure 5(a)**) the user can obtain the colorful picture (**Figure 5(b)**) using the method proposed by Levin et al., while users can still obtain the perfect visual effect (**Figure 5(e)**) only by drawing several strokes (**Figure 5(c)**) using the method proposed by Luan et al. **Figure 6** shows that when the user want to colorize a picture (**Figure 6(a)**), the method proposed by Xu et al. can produce better result (**Figure 6(b)**) where the background color and the color of the fruits look closer to the desiring color than the result produced by Levin et al. (**Figure 6(c)**). Therefore, the user can obtain the same result or a better result with a few strokes compared to the methods that need a lot of user strokes and thus the colorization process becomes more convenient for the user.

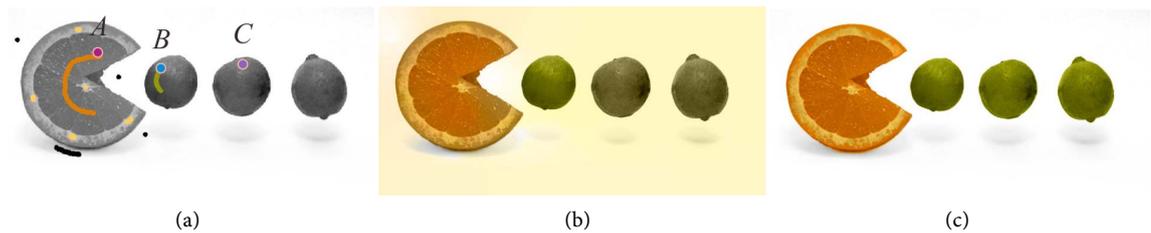
## 4. Colorization on Sketch

### 4.1. Overview

There is a significant difference between grayscale images and sketch images: sketch images have no grayscale information. While many scribble-based colorization algorithms used in colorizing grayscale images were invented after Levin et al., few algorithms can achieve desirable results when they are employed in colorizing sketch images.



**Figure 5.** Comparison of Colorization Results (Luan et al., 2007). (a) Stokes needed by previous methods; (b) Colorization result using Levin et al. (2004); (c) Stokes needed by Luan et al. (2007); (d) Color mapping; (e) Colorization result using Luan et al. (2007).



**Figure 6.** Comparison of Colorization Results (Xu et al., 2013). (a) Strokes on image; (b) Colorization result using Levin et al. (2004); (c) Colorization result using Xu et al. (2013).

Qu et al. first designed a method that can propagate colors through pattern-continuous and intensity-continuous regions in manga (Qu et al., 2006). Regions with similar pattern features and with open boundaries can be segmented intelligently. After the segmentation, several colorization techniques can be employed in filling in colors.

In the context of colorizing black-and-white cartoons, however, Sykora et al. pointed out the limitations in Manga Colorization and briefly discussed an ideal painting tool instead (Sýkora et al., 2009). An ideal painting tool may have four major properties as following: 1) It tends to achieve the “largest” colorized area by seeking an optimal boundary, regardless of inner holes and gappy outlines; 2) In a locality, the color is continuous and determined by the nearby scribbles; 3) Soft scribbles (imprecisely-placed strokes, with very few parts outside as well as the majority lying in the interior) are preferred rather than the subtle strokes inside the region; 4) The anti-aliasing effect in the original image is preserved (to address the problem of colorizing regions with vague boundaries). A new black-and-white colorization algorithm is carefully designed pursuing the four ideal properties, and it significantly enhanced the convenience and achieved the very same visual effect with less precise strokes (Figure 7).

## 4.2. Theory

### 4.2.1. Manga Colorization

Qu et al. use level set method to describe the evolving curve in order to implement



**Figure 7.** From left to right: scribbled image in Qu et al., scribbled image in Qu et al. after highlighting all user strokes, scribbled image in Sykora et al., and colorization result in Qu et al. and in Sykora et al., respectively. ©Yukito Kishiro/Shueisha.

segmentation. By using appropriate formulas to express the propagating speed, this method can achieve desirable segmentation results. Various techniques of colorization will then be applied in color-filling after segmentation.

According to (Qu et al., 2006), manga colorization includes two parts: segmentation and color-filling.

Three methods used in color-filling were mentioned by Qu et al.: color replacement, stroke-preserving colorization, and pattern to shading. Color replacement method replaces the black or white color on original manga image by the color on the user's scribble. Stroke-preserving colorization preserves the original pattern during colorization and pattern to shading method transforms the pattern into smooth color shading.

As for segmentation, Qu et al. used the level set method to describe the evolving curve. According to Hamilton-Jacob equation, the following equation can be raised:

$$\frac{\partial \phi}{\partial t} = -F |\nabla \phi| \quad (9)$$

where  $\phi$  is the implicit 3D function,  $t$  is the evolving time, and  $F$  denotes the moving speed of the evolving front. For segmentation problems,  $F$  can be replaced by the following equation:

$$F = h * (F_A + F_G) \quad (10)$$

where  $F_A$  is normally a constant and  $F_G$  depends on the geometry of the curve's evolving front.  $h$  is a term used to abort the curve evolution, and has different expressions in pattern-continuous regions and intensity-continuous regions.

- *Pattern-continuous regions*

In the pattern-continuous regions,  $h$  can be expressed by the following formula:

$$h_p(x, y) = \frac{1}{1 + |D(T_{\text{user}}, T_{\text{front}}(x, y))|} \quad (11)$$

where  $T_{\text{front}}(x, y)$  and  $T_{\text{user}}$  are the pattern feature on the evolving front and that on the user scribbles, respectively. Function  $D$  measures the difference

between  $T_{\text{user}}$  and  $T_{\text{front}}(x, y)$ . Pattern feature  $T$  can be expressed by the statistical feature in Gabor wavelet domain: after implementing Gabor wavelet transforming, the mean value  $\mu_{m,n}$  and the value of standard deviation  $\sigma_{m,n}$  will be calculated in order to express the feature vector:

$$T = [\mu_{0,0} \sigma_{0,0} \mu_{0,1} \cdots \mu_{3,5} \sigma_{3,5}] \quad (12)$$

- *Intensity-continuous regions*

In the intensity-continuous regions,  $h$  can be expressed as:

$$h_l(x, y) = \frac{1}{1 + |\nabla(G_\sigma \otimes I(x, y))|} \quad (13)$$

where  $G_\sigma$  is the Gaussian smoothing filter.  $|\nabla(G_\sigma \otimes I(x, y))|$  is normally zero except at the places where abrupt image gradient change happens.

To implement leak-proofing, a new equation of  $F$  is denoted as:

$$F = h_l(F_A + F_l + F_G) \quad (14)$$

$F_l$  can be expressed by:

$$F_l(x, y) = -F_A R \left( \frac{|\nabla G_\sigma \otimes I(x, y)| - M_2}{M_1 - M_2 - \delta} \right) \quad (15)$$

where  $M_1$  and  $M_2$  are the maximum and minimum values of  $|\nabla(G_\sigma \otimes I(x, y))|$ . Relaxation factor  $\delta \in [0, M_1 - M_2]$  and the result of function  $R$  is between 0 (include) and 1 (include).

#### 4.2.2. Lazybrush

Sykora et al. converted the four requirements they proposed into an energy minimization framework, by which a desirable result was obtained in colorizing black-and-white cartoons.

- *Energy Minimization Framework*

Given a grayscale image  $I$ , the energy function of a colorization scheme  $S$  is denoted by:

$$E(S) = \sum_{p \in I} \sum_{q \in N(p)} E_{p,q}(S_p, S_q) + \sum_{p \in I} E_p(S_p) \quad (16)$$

where  $S_p$  is the color assigned to pixel  $p$  in the scheme,  $E_{p,q}$  is the energy of color discontinuity between two neighboring pixels  $p$  and  $q$ , and  $E_p$  is the energy of assigning color  $S_p$  to pixel  $p$ . Minimizing the energy function, considering both the effect of the pixel-wise color assignment and the chrominance correlation of two neighboring pixels, gives us an optimal scheme of colorizing black-and-white sketches, satisfying the proposed properties (or requirements).

The reason for introducing  $E_{p,q}$  is to manage color discontinuities. One simple observation is that though the original image is a typical black-and-white drawing, there may still exist a lot of vague boundaries. Another observation is that color discontinuities appear in narrow regions with low-intensity pixels. Therefore, contrast enhancement process is performed first, where the outlines

are extracted after applying a Laplacian of Gaussian ( $L \circ G$ ) filter on the source image. Subsequently, the energy of color discontinuity is achieved on the basis of the intensities of the pixels in the processed image.

Term  $E_p$  measures the effect of the pixel-wise color assignment.  $E_p$  takes on three distinct values corresponding to three different conditions of pixels, namely not scribbled, under hard scribbles and under soft scribbles.

- *Multiway Cut Problem*

Minimizing the proposed energy function can be reduced to solving a multiway cut problem in that the energy of color discontinuity mainly depends on the luminance rather than the chrominance of the strokes (Potts, 1952; Boykov et al., 1998).

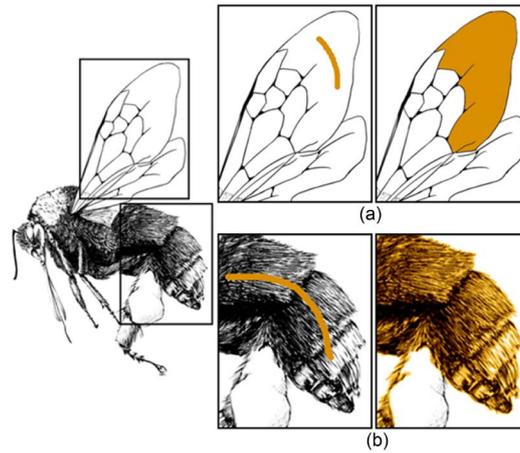
Given an undirected graph  $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$  ( $\mathcal{V}$  is a set of vertices consisting of both colored and uncolored vertices, and  $\mathcal{E}$  is a set of edges each assigned with a weight), the multiway cut problem is to find a scheme of removing edges to properly partition the graph according to color labels (colored pixels), while obtaining a resulting set of removed edges with the minimum sum of weights at them.

Since multiway cut problem with three or more terminals is currently intractable in a polynomial time (Dahlhaus et al., 1992), the resulting schemes can only guarantee a optimality within a certain factor (Dahlhaus et al., 1992; Boykov et al., 2001; Karger et al., 2004). Sykora et al. proposed a hierarchical method of greedily and gradually dividing the original problem into min-cut subproblems and pruning trivial cases with only one terminal. Compared with the widely-applied  $\alpha$ -expansion algorithm (Boykov et al., 2001), this algorithm is significantly more efficient and therefore is more suitable to be implemented in this interactive colorization setting, though it only approximates the optimal solution  $r$  within a comparatively larger factor (i.e. a lower precision).

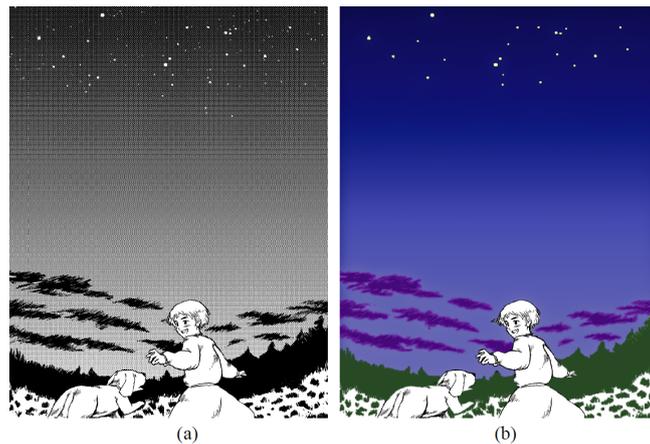
### 4.3. Experiment

Compared with previous intensity-based colorization methods, the results produced by Qu et al. (2006) is much more satisfying. Algorithm proposed by Qu et al. can propagate intelligently through pattern-continuous regions (Figure 8(b)) and intensity-continuous regions (Figure 8(a)). after the curve evolution, three kinds of methods can be employed in colorization: color replacement (Figure 8(a)), stroke-preserving colorization (Figure 8(b)), and pattern to shading (Figure 9).

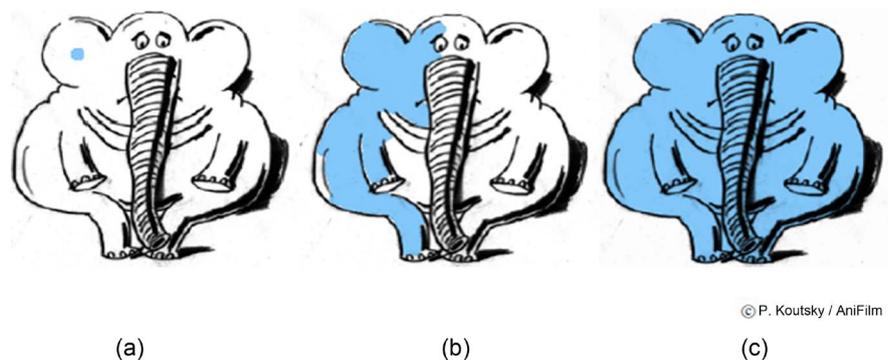
However, there are still many limitations in Qu et al. where Sykora et al. significantly improves. When given an input with very few strokes in the interior (Figure 10), LazyBrush is able to determine and fill in an optimal boundary of the region, while the algorithm in Manga Colorization gets stuck in some un-intended regions and may require extra strokes to achieve the same result. Besides, in order to achieve a desirable result, more tedious subtle strokes are also needed in Manga Colorization, while many im-precisely-placed strokes are tolerable in LazyBrush (Figure 7).



**Figure 8.** (a) Colorization in intensity-continuous regions, using color replacement method; (b) Colorization in pattern-continuous regions, using stoke-preserving method.



**Figure 9.** Colorization in pattern-continuous regions, using pattern to shading method.



**Figure 10.** Comparison of the visual effect between different algorithms. (a) Input; (b) Result presented by Qu et al. (2006); (c) Result presented by Sýkora et al. (2009).

## 5. Colorization with Neural Network

With the development of deep learning, many studies of different areas incorporate deep learning networks into their frameworks. Colorization, as one of those areas, can be improved by the help of deep learning networks.

## 5.1. Method

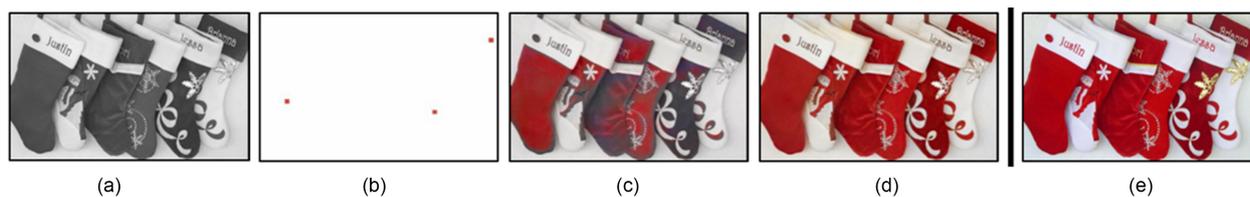
Zhang et al. (2017) proposed to train the Convolutional Neural Network (CNN) with many images and thus applied this network to colorize new grayscale images. Specifically, Zhang et al. implemented the U-Net structure (Ronneberger et al., 2015) to realize colorization, designing feature extraction part, global input part, fusion part and reconstruction part. However, during the process of training the CNN, it's very difficult to collect user input data. To overcome this difficulty, they obtained the simulating user inputs by sampling several points on the colorful images. The parameter of CNN is calculated by minimizing the cost function:

$$\theta^* = \arg \min_{\theta} E_{X,U,Y-D} [\Gamma(F(X,U;\theta), Y)] \quad (17)$$

where  $\theta^*$  is the parameter that minimizes the loss function;  $X$  is the grayscale image;  $U$  is the user input;  $Y$  is the color output;  $F$  is the CNN network with the parameter  $\theta$ ;  $\Gamma$  is the function describing the closeness between the true value and the output value.

## 5.2. Performance

After experiments, Figure 11 (Zhang et al., 2017) shows that this method proposed by Zhang et al. can work better than the method proposed by Levin et al. when user only input sparse points and Figure 12 (Zhang et al., 2017) presents the network can work well even when user wants to colorize unusual colors.



**Figure 11.** Comparison of different colorization results with the input of sparse user input. (a) Grayscale image input; (b) User input; (c) Result with the method proposed by Levin et al.; (d) Result with CNN; (e) Truth image.



**Figure 12.** Output with unusual user input. (a) Unusual user input; (b) Result with CNN; (c) Truth image.

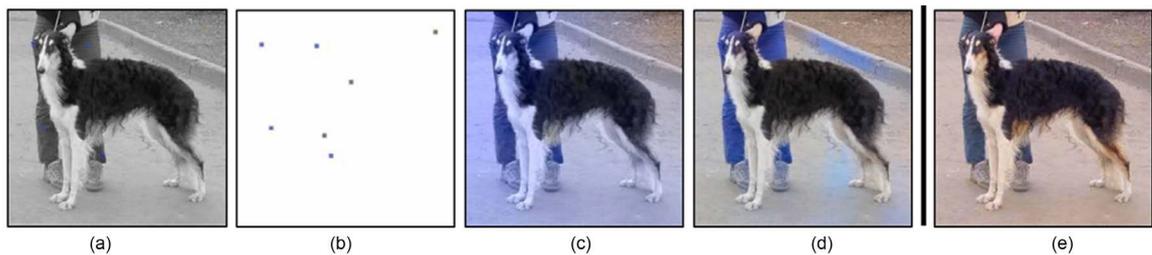
### 5.3. Limitation

However, **Figure 13** (Zhang et al., 2017) shows that when we want to colorize the jeans with blue, the background is unfortunately colorized with blue. Therefore, sometimes, the foreground color will influence the background color. In this case, we need more user points to fix this problem.

### 6. Conclusion

In this survey paper, we introduce many algorithms about scribble-based colorization published in recent years, divided into three types: 1) colorization on grayscale image; 2) colorization on sketch; 3) colorization with networks. For each area, we present their principles, mathematical methods and experiments.

Comparing different algorithms in grayscale image colorization and sketch colorization, we found out that different mathematical tools were applied by researchers to implement colorization. Each of these tools has its different performance on the respect of efficiency, convenience, leak-proof and continuity (and also anti-aliasing and special effects) (**Table 1** and **Table 2**): “Efficiency” is



**Figure 13.** Comparison of different colorization results with the input of sparse user input. (a) Grayscale image input; (b) User input; (c) Result with the method proposed by Levin et al.; (d) Result with CNN; (e) Truth image.

**Table 1.** Grayscale image colorization.

	Efficiency	Convenience	Leak-proof	Color continuity
Levin et al. (2004)	★	★	★	★
Huang et al. (2005)	★★	★	★	★★
Yatziv & Sapiro (2006)	★★★★★	★★	★★★	★★★★★
Luan et al. (2007)	★★	★★★	★★★★	★★
Xu et al. (2013)	★★	★★★★★	★★★★	★★★★

**Table 2.** Sketch colorization.

	Convenience	Leak-proof	Color continuity	Anti-aliasing	Special effect
Levin et al. (2004)	★★	★	★★	×	×
Qu et al. (2006)	★★★	★★★★	★★★★	×	√
Sýkora et al. (2009)	★★★★★	★★★★	★★★★	√	×

“×” represents bad performance and “√” represents good performance; “★” represents the degree of performance, so more stars mean better performance.

measured by the running time under the same settings; “Convenience” is a subjective metric for users and algorithms are ranked according to the extent of required user intervention; “Leak-proof” measures the algorithms’ abilities of avoiding color leaking; “Color-continuity” demonstrates the “smoothness” of colors in images which is also an important metric of visual effects; For Sketch Colorization, the ability to recognize and utilize anti-aliasing of lines in the original image (“Anti-aliasing”), and ability to preserve and colorize the special effects (like gradients) are also displayed in the table. Also, it should be noted that these tables may inexplicitly display some subjectivities and “five stars” here just conveys that the algorithm graded achieves a comparatively better (or the best among all shown ones) performance.

Few of the surveyed algorithms are capable of achieving perfect visual effects while minimizing the amount of user-intervention, because it is very difficult to design a specific mathematical model to cover all situations. In light of the absence of the methods that can achieve perfect visual effects efficiently with only several strokes, it still needs more works to optimize the methods mentioned above.

In addition, with the development of deep learning, colorization methods using neural network are more pervasive. Unfortunately, the parameters of a network are very difficult to be determined because people still know very little about relations between the parameters in neural network and actual features in an image. Therefore, the architecture of a neural network should be carefully designed. However, these methods combined with networks can colorize complex images with some sense of intelligence, which is very useful for illustrators. In conclusion, colorization methods combined with advanced artificial neural networks will be very promising in the future.

## Acknowledgements

We would like to thank Prof. Barsky for instructing our investigation and we also would like to thank Yang and Saba for helping us with paper writing.

## Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

## References

- Boykov, Y., Veksler, O., & Zabih, R. (1998). *1998 IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (pp. 648-655). Piscataway, NJ: IEEE.
- Boykov, Y., Veksler, O., & Zabih, R. (2001). Fast Approximate Energy Minimization via Graph Cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *23*, 1222-1239. <https://doi.org/10.1109/34.969114>
- Dahlhaus, E., Johnson, D. S., Papadimitriou, C. H., Seymour, P. D., & Yannakakis, M. (1992). *Proceedings of the Twenty-Fourth Annual ACM Symposium on Theory of Computing* (pp. 241-251). New York City: ACM.

- Huang, Y. C., Tung, Y. S., Chen, J. C., Wang, S. W., & Wu, J. L. (2005). *Proceedings of the 13th Annual ACM International Conference on Multimedia* (pp. 351-354). New York City: ACM. <https://doi.org/10.1145/1101149.1101223>
- Karger, D. R., Klein, P., Stein, C., Thorup, M., & Young, N. E. (2004). Rounding Algorithms for a Geometric Embedding of Minimum Multiway Cut. *Mathematics of Operations Research*, 29, 436-461. <https://doi.org/10.1287/moor.1030.0086>
- Levin, A., Lischinski, D., & Weiss, Y. (2004). Colorization Using Optimization. *ACM Transactions on Graphics (Tog)*, 23, 689-694. <https://doi.org/10.1145/1015706.1015780>
- Lischinski, D., Farbman, Z., Uyttendaele, M., & Szeliski, R. (2006). Interactive Local Adjustment of Tonal Values. *ACM Transactions on Graphics (TOG)*, 25, 646-653. <https://doi.org/10.1145/1141911.1141936>
- Luan, Q., Wen, F., Cohen-Or, D., Liang, L., Xu, Y. Q., & Shum, H. Y. (2007). *Proceedings of the 18th Eurographics Conference on Rendering Techniques* (pp. 309-320). Eurographics Association.
- Potts, R. B. (1952). Some Generalized Order-Disorder Transformations. *Mathematical Proceedings of the Cambridge Philosophical Society*, 48, 106-109. <https://doi.org/10.1017/S0305004100027419>
- Qu, Y., Wong, T. T., & Heng, P. A. (2006). Manga Colorization. *ACM Transactions on Graphics (TOG)*, 25, 1214-1220. <https://doi.org/10.1145/1141911.1142017>
- Ronneberger, O., Fischer, P., & Brox, T. (2015). *International Conference on Medical Image Computing and Computer-Assisted Intervention* (pp. 234-241). Cham: Springer.
- Sýkora, D., Dingliana, J., & Collins, S. (2009). Lazybrush: Flexible Painting Tool for Hand-Drawn Cartoons. *Computer Graphics Forum*, 28, 599-608. <https://doi.org/10.1111/j.1467-8659.2009.01400.x>
- Xu, L., Yan, Q., & Jia, J. (2013). A Sparse Control Model for Image and Video Editing. *ACM Transactions on Graphics (TOG)*, 32, 197. <https://doi.org/10.1145/2508363.2508404>
- Yatziv, L., & Sapiro, G. (2006). *Fast Image and Video Colorization Using Chrominance Blending*. *IEEE Transactions on Image Processing*, 15, 1120-1129.
- Zhang, R., Zhu, J. Y., Isola, P., Geng, X., Lin, A. S., Yu, T., & Efros, A. A. (2017). *Real-Time User-Guided Image Colorization with Learned Deep Priors*. arXiv Preprint arXiv:1705.02999.