EUROGRAPHICS 2019 / P. Alliez and F. Pellacini
(Guest Editors)

*Volume 38 (2019), Number 2*

# Generating Color Scribble Images using Multi-layered Monochromatic Strokes Dithering

Yi-Hsiang Lo[1] Ruen-Rone Lee[2] Hung-Kuo Chu[1]

[1]National Tsing Hua University, Taiwan.
[2]Industrial Technology Research Institute, Taiwan.

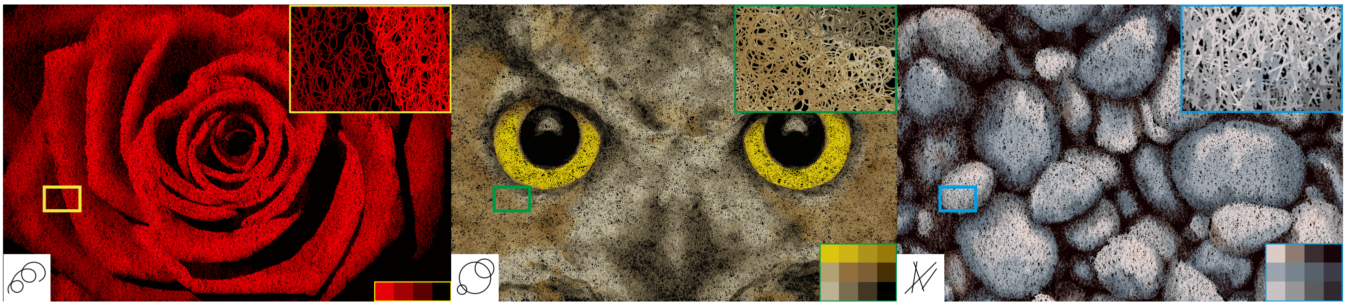


**Figure 1:** Color scribble images generated using our system with limited color palettes shown on the bottom right corners. The insets on the top right corners show the detailed scribble patterns with different stroke styles as shown on the lower left corners (from left to right: circular scribbles [CLLC15], circles, and line segments). **Please zoom into the image using the digital version to further observe how different layers of monochromatic scribbles are composed to depict the original color images.**

**Abstract**

*Color scribbling is a unique form of illustration where artists use compact, overlapping, and monochromatic scribbles at microscopic scale to create astonishing colorful images at macroscopic scale. The creation process is skill-demanded and time-consuming, which typically involves drawing monochromatic scribbles layer-by-layer to depict true-color subjects using a limited color palette delicately. In this work, we present a novel computational framework for automatic generation of color scribble images from arbitrary raster images. The core contribution of our work lies in a novel color dithering model tailor-made for synthesizing a smooth color appearance using multiple layers of overlapped monochromatic strokes. Specifically, our system reconstructs the appearance of the input image by (i) generating layers of monochromatic scribbles based on a limited color palette derived from input image, and (ii) optimizing the drawing sequence among layers to minimize the visual color dissimilarity between dithered image and original image as well as the color banding artifacts. We demonstrate the effectiveness and robustness of our algorithm with various convincing results synthesized from a variety of input images with different stroke patterns. The experimental study further shows that our approach faithfully captures the scribble style and the color presentation at respectively microscopic and macroscopic scales, which is otherwise difficult for state-of-the-art methods.*

**CCS Concepts**
• ***Computing methodologies*** $\rightarrow$ ***Non-photorealistic rendering;***

## 1. Introduction

Scribbling is a kind of illustrative drawing where people use seemingly random and careless scribble lines to depict images or conceptual designs. For grayscale images, artists typically use monochromatic strokes (e.g., black) to depict the original tones by controlling the density and path of scribble lines when doodling (see the circular scribble artworks shown in [CLLC15]). Such unique drawing skill is further elevated to a higher level when dealing with true color images where the compact, multi-layered scribbles are drawn and overlapped deliberately at microscopic scale to reproduce the original color depth at a macroscopic scale (see Fig-

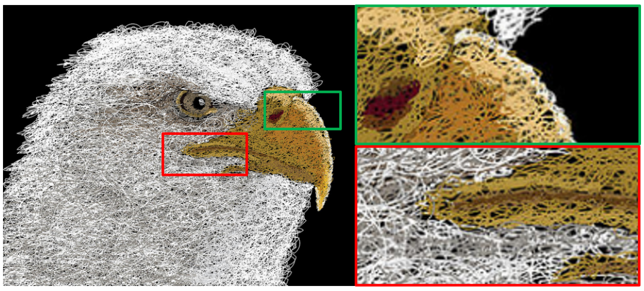


**Figure 2:** An example of color scribble art (©Nathan Shegrud)

ure 2). Note that the above mentioned artistic style is very different from another commonly seen color pencil art created by the artist John P. Smolko [Smo13], who skillfully utilizes scattered color scribbles with various thickness and patterns to add abstract expressions on the original images.

As an art form, color scribble image shares a common spirit with traditional watercolor or oil painting to faithfully depict colorful images using only a limited number of color pigments. However, unlike watercolor or oil painting where stroke color can be precisely determined via mixing base pigments in the palette before painting, color scribbling demands delicate controls over drawing multiple layers of monochromatic scribbles at microscopic scale in a manner that the original color depth can be perceived at macroscopic scale (see Figure 2). The mechanism of using a limited color palette to hallucinate full-color spectrum is also referred to *dithering* in digital photography and image processing. Conventional dithering methods employ error diffusion in the separated color channels (e.g., CMYK) to propagate the color quantization error from a pixel to its neighbors [FS76]. Nevertheless, implementing the color dithering via color scribbles with complex shapes and structures is nontrivial and requires trial and error during the drawing. Thus, the creation of color scribble images remains a tedious and time-consuming process even for skillful artists.

In this work, we present a novel computational framework for automatic generation of color scribble images from arbitrary raster images. The core component of our system is a novel color dithering model tailor-made for synthesizing a smooth color ramp via overlapping multiple layers of monochromatic strokes. The design principle behind our model is driven by typical color dithering methods that focus on modeling the following two key problems: (i) how to reproduce a smooth color ramp using a small set of base colors, and (ii) how to eliminate the *color banding* artifacts due to the quantization error. In response to these questions, we propose a *visual color evaluation function* to predict a target color in a local region with multiple layers of overlapped monochromatic scribbles. The objective function is formulated to measure *color dissimilarity* between the synthesized and original color ramp, and meanwhile capture the *color banding* artifact between adjacent ramp colors. The dithering problem is then solved by optimizing the assignment of base colors to the constituted scribbling layers of individual quantized ramp colors such that the reconstructed color ramp has minimal visual distortion.

Given an input image, we generate the corresponding color scribble image as follows. Our system starts with computing a color palette by selecting representative base colors among several quantized color ramps derived from image segmentation and quantization. Next, we extend the color dithering model from 1D ramp to 2D image by considering the spatial relationships among adjacent image segments that are associated to the same quantized color ramp and compute the optimal base color assignment. The system then employs a scribble generator to synthesize monochromatic scribbles at each layer using the assigned base color. The drawing sequence among different scribble layers is carefully scheduled to further enhance the smoothness across image regions. We evaluated our system on a wide variety of input images to generate various color scribble images. Figure 1 shows some typical examples generated using our system with different scribble patterns such as circular scribbles [CLLC15], circles with varying size, and line segments with random orientation. Experimental study shows that our approach outperforms the state-of-the-art methods, including texture synthesis and learning-based image-to-image translation, particularly in preserving the acute visual stylization quality at a microscopic scale as well as a color presentation at a macroscopic scale.

In summary, our work makes two major contributions:

- An automatic framework to synthesize color scribble images from arbitrary raster images. To the best of our knowledge, our work is the first attempt in the field.
- A novel layer-based color dithering model tailor-made for scribble patterns to achieve image dithering with minimal visual color dissimilarity and color banding artifacts.

## 2. Related Works

**Computational digital arts.** The topic of generating images with particular art form has been extensively explored in the field of non-photorealistic rendering. Among those previous researches, our work is closely related to the image stylization in the context of halftoning [MP92, LAG98, MV02, PQW*08, BH13], stippling [Sec02, JH05, ALMPHS10], hatching [PHWF01, WPFH02], line drawing [KB05, WT11, WT13, TFL13], and scribbling [CLLC15]. Halftoning and stippling are commonly used techniques for approximating continuous tone using discrete dots with varying size and sampling density. Likewise, hatching, line drawing, and scribbling depict the shape and tone of an image through a well-defined parametric model to synthesize stroke patterns with varying spacing, thickness, and different degrees of overlapping. However, these methods mainly focus on preserving the structure and tone of grayscale images. Although Chiu *et al.* [CLLC15] presented an extension of their approach to color images by applying the circular scribbles generator to individual channels (e.g., RGB or CMYK), such a naïve application can not faithfully reproduce the original color appearance and generate satisfactory results. In this work, we explore a completely different style that reconstructs the appearance of color images based on a color dithering mechanism, which superimposes multiple layers of monochromatic scribbles.

**Color quantization and dithering.** Color quantization and dithering are commonly used techniques to compress a true-color image into a compact representation using a small number of colors [OH99, VF03]. Wu [Wu92] presented a color-space-partitioning

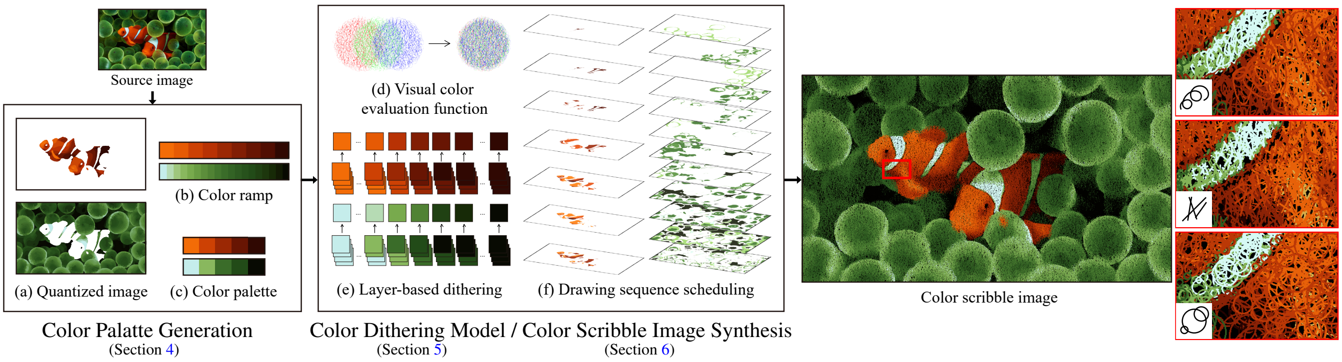



**Figure 3: Method overview.** Given a source image, our system starts with generating a color palette (c) where the base colors are selected out of quantized color ramps (b) that are derived from image segmentation and quantization (a). Then, a novel layer-based color dithering model (e), guided by a proposed visual color evaluation function (d), is applied to synthesize the target colors in original ramps by superimposing layers of monochromatic scribbles whose colors are selected from the extracted color palette. The drawing sequence (f) of these color scribble layers is further computed to reconstruct the original color ramps with minimal color banding artifacts. The final result is obtained by combining the color scribbles synthesized from different color ramps. The rightmost three renderings show that our method is compatible to various stroke patterns.

quantization algorithm along a statistical principal axis of pixel colors. Error diffusion methods [FS76, Ost01] were used to propagate color quantization error from a pixel to its neighbors to obtain smooth results. Other approaches employed global optimization [PHK*00, HXM*16] to simultaneously perform quantization and dithering on the true-color images. Instead of working at pixel level, color scribble image implements the color dithering by superimposing multiple layers of monochromatic strokes with complex shapes and structures. This poses a nontrivial effort to reconstruct the original color image without introducing significant visual distortion.

**Image decomposition.** Decomposing a color image into segments or layers based on the color or spatial information of constituting pixels plays an important role in early-stage step of many advanced applications such as image editing and color transfer [TJT07, TLG17, AASP17]. For example, the soft color image segmentation generates segments with overlapped portions which contain not only colors but also opacity information for better composition or editing. The pixel level opacity does help in the fine image operations such as image matting or recoloring. However, unlike the pixel-based image synthesis, for stroke-based scribbles with unpredictable scribble intersections during doodling, per-pixel opacity is not practical in the color scribble synthesis. While these previous works can be adapted to our system with minor changes, we found that a simple pixels grouping based on the quantized colors is enough and produce satisfactory results in our context.

**Style transfer.** Generating art-like color scribble images is also related to the research topic of transferring intrinsic style elements across images. Conventional style transfer approaches such as texture synthesis and texture transfer [EF01, HJO*01, PHWF01] are based on the idea of copying local patches from the reference image and pasting these patches on the target image. Recent advances made with deep learning inspired a line of works toward developing an end-to-end neural network model to achieve style transfer in an either supervised or semi-supervised manner. For example, Gatys *et al.* [GEB16] and Liao *et al.* [LYY*17] use a convolutional neural network (CNN) to learn a given artistic style and transfer both the style and color to a target image. Other attempts trained a generative adversarial network (GAN) with either paired or unpaired data to achieve robust image-to-image translation [IZZE17, ZPIE17]. Although previous methods have presented impressive results in transferring styles from image to image, they failed to generate satisfactory outputs when dealing with color scribble images. They all lacked the capability of preserving the shapes and structures encoded in the reference image.

## 3. Overview

An overview of the proposed framework is illustrated in Figure 3. Given an input source image, our system starts with generating a color palette where the base colors of quantized color ramps are derived from image segmentation and quantization (Section 4). To reconstruct the appearance of original color ramps, we adopt a novel color dithering model that superimposes multiple layers of monochromatic scribbles with optimal colors selected from the extracted color palette (Section 5). Then, a drawing sequence among different scribble layers is carefully scheduled to smooth out the color transition between adjacent image regions and hence reduce the color banding artifacts induced by quantization error (Section 6). The final color scribble image is obtained by combining the color scribbles generated with respect to image regions associated with different color ramps.

## 4. Color Palette Generation

To depict a true color image using only a compact set of color scribbles, a color quantization process is needed to derive a color palette before the image can be adapted to our dithering model. There are also several well-known color quantization approaches such as median cut [Hec82], octree [GP88], or local mapping and thresholding [WOG06], which dedicate to minimize the quantization error.

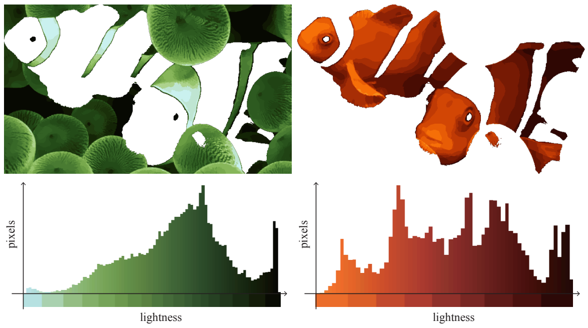



**Figure 4:** An example of segmenting the source image into two color ramps (upper row) and the corresponding histograms with respect to their lightness values (bottom row).

However, performing monochromatic stroke dithering requires not only colors to be quantized into several clusters, but also the colors within a cluster are sorted as a ramp for ease of blending and reproducing the original color depth in our dithering process. In the following paragraph, we elaborate a simple yet effective approach to derive a color palette from an input image.

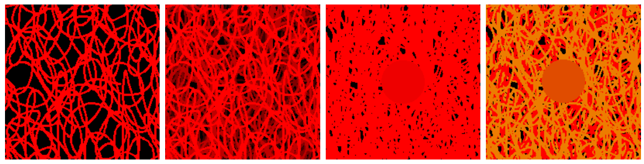
We first convert the input image into HSL color space and apply $K$-means clustering [Llo82] on the hue (H) and saturation (S) components to classify pixels into clusters. Pixels within each cluster are sorted by their lightness (L) and divided into $m$ groups of equal number of pixels (see Figure 4). The quantized color ramp of the $k^{th}$ cluster can be obtained by averaging the pixel colors of each group and denoted as $R_k = \{C_1^k, C_2^k, ..., C_m^k\}$. For each $R_k$, the base colors are determined by a uniform sampling of $n$ quantized colors among $R_k$, including $C_1^k$ and $C_m^k$. The final color palette is then defined as the union of base colors among all color ramps. In our experiment, the number of clusters is determined manually according to the complexity of input images. For the color ramp quantization, we found the empirical settings of $2^5 \leq m \leq 2^6$ and $4 \leq n \leq 6$ with 4 layers of scribbles are sufficient to generate delicate results (see results in Figure 12). The evaluation regarding the results generated using different numbers of base colors can be found in Section 7.3.

## 5. Color Dithering Model

In order to synthesize the color image using color scribbles with limited base colors, a color dithering model is introduced to restore the original color depth in a color ramp. Traditional point-based dithering approaches, using pixels or dots, have difficulties in preserving stroke styles which have long and continuous features such as lines or scribbles. We present a novel layer-based color scribble dithering model to synthesize the color ramp of an input image with minimum color dissimilarity and color banding artifact.

### 5.1. Scribble Pattern Synthesis

Our system implements three kinds of color scribble patterns, the circular scribble, circles, and line segments. For the synthesis of circular scribbles, we utilize the generator proposed by [CLLC15] with the parameters setting of sample radius ($r_{sample} = 24$), center velocity ($V_c = 2.5$), and scribble radius ($r_{max} = 40$) on a $4000 \times 3000$ canvas. Please refer to [CLLC15] for detailed implementation of the generator. As for the circles and line segments, we sample along the path of circular scribbles and draw a circle with a random radius $[5, 45]$, and draw a line with random orientation $[0, \pi]$ and length $[10, 60]$. For the sake of simplicity, we use the circular scribbles to illustrate the core algorithm and demonstrate the application of different scribble patterns in the visual results (see Figure 1 and Figure 12). The evaluation regarding the results generated using different configurations can be found in Section 7.3.



(a) (b) (c) (d)

**Figure 5:** (a) One layer of scribbles. (b) Four layers of scribbles (the layers are drawn from the bottom to top in dark red to bright red respectively). (c) Four layers of red scribbles in RGB color (255, 0, 0) introduce a visual color of (238, 0, 0) as shown in the filtered center area. (d) Four layers of scribbles (red, orange, red, and orange, respectively) introduce a visual color of (224, 76, 0).

### 5.2. Visual Color Evaluation Function

By drawing multiple layers of color scribbles, it may result in different visual colors due to the coverage of color scribbles for each layer in a unit local region is different. To evaluate the final perceived visual color, the coverage ratio $r(I)$ between the number of scribble pixels $N(I)$ to the number of pixels in a local canvas region $I$ is defined as

$$r(I) = \frac{N(I)}{area(I)} \tag{1}$$

Based on the coverage ratio, the visual color $V(I)$ of a scribble with color $C$ over a local region $I$ is defined by

$$V(I) = r(I) \times C \tag{2}$$

We regard each pass of scribble strokes drawn on the local region with a color as a *layer*. In Equation 2, the layer of scribble strokes with color $C$ has contributed to the visual color $V(I)$ with coverage ratio $r(I)$. Figure 5(a) shows a result of drawing one layer of scribbles on a black canvas. The human visual perception system will automatically blend the scribble color with canvas color such that it appears to be a color of dark red at a distance.

For simplicity, we will assume that when drawing a layer of color scribbles, it will directly supersede the pixel colors beneath it, as can be seen in common digital scribble artworks such as the one shown in Figure 2. Thus, for multiple layers of color scribbles, the final visual color $C'$ can be defined as a *Visual Color Evaluation Function*:

$$C' = \sum_{i=1}^{L} (\omega_i \times C_i) + (1 - \sum_{i=1}^{L} \omega_i) \times C_c \tag{3}$$

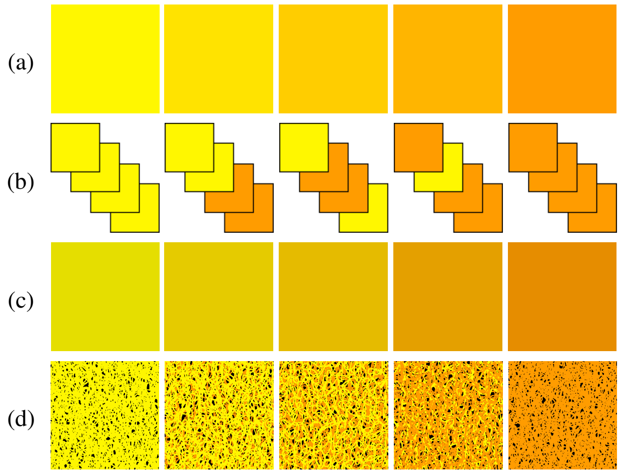



**Figure 6:** Illustration of color reproduction using layers of two base colors (yellow and orange). (a) target segment colors $C_j$; (b) combinations of two base colors $C_{j,i}$ in four layers; (c) predicted visual colors $C'_j$ (Equation 3); (d) color scribble dithering results with four layers of scribbles.

The formula is similar to the Neugebauer equations [Neu37] in predicting the printed color of a combination of CMYK color halftone inks. Scribbles are drawn from the first layer (the bottom layer) to the $L^{th}$ layer (the top layer) in order. In Equation 3, $C_i$ denotes the scribble color used in the $i^{th}$ layer and $C_c$ is the canvas color. The weighting factor $\omega_i$ represents the coverage ratio of scribbles for the $i^{th}$ layer that remain visible after $L$ layers of scribbles are drawn.

In order to control the colors synthesized by our dithering model, we assume that the coverage ratio of drawing a layer of color scribbles is a constant $k$. For a large coverage ratio $k$, it is difficult to synthesize the expected colors since the top layer color will dominate the final color. However, for a small coverage ratio $k$, it requires much more layers of color scribbles to finally approach the desired color. Thus, we set the coverage ratio of drawing a layer of color scribbles to 0.5 (i.e., $k = 0.5$) for balancing the number of layers used as well as the number of colors that can be synthesized. We manually adjusted the parameters of scribble patterns and obtained an empirical setting (see Section 5.1) such that a layer of scribble strokes within a unit region can achieve the approximate coverage ratio 0.5. Based on the defined coverage ratio $k$, it is clear that we have $\omega_L = k$ for the top layer (the $L^{th}$ layer) and $\omega_{L-1} = k(1-k)$ for the next layer (the $(L-1)^{th}$ layer). That is, for calculating the coverage ratio of visible scribble pixels on the specific layer $i$, we have $\omega_i = k(1-k)^{L-i}$, where $1 \le i \le L$.

Figure 5(b), (c), and (d) demonstrate examples of drawing four layers of specific color scribbles onto a black canvas. Based on the weighting factors $\omega_i$ discussed above, the resulting colors derived from Equation 3 in a local region are close to the visual colors measured by using a box filter in the center regions, respectively.

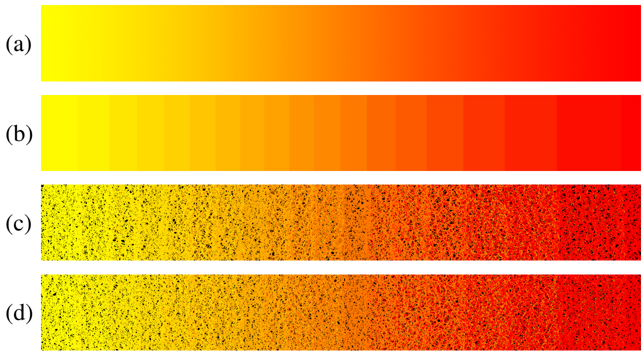



**Figure 7:** A color band with smooth transition from yellow to red is shown in (a). The colors are sorted and reduced to 19 ramp colors as shown in (b). The color scribble results using 5 palette colors before and after applying smoothness optimization are shown in (c) and (d) respectively.

### 5.3. Color Reproduction

Color reproduction is used to synthesize colors in a color ramp by a combination of designated base colors. As mentioned in Section 4, a set of color ramps is derived by segmenting the image into clusters. Assume that a cluster of an image with its associated color ramp is quantized and sorted into $q$ color segments $Q_j$ with $C_j$ as the respective quantized color for $1 \le j \le q$. Then, we can approximate the colors $C_j$ by drawing $L$ layers of color scribbles with two adjacent base colors $C_a$ and $C_b$, where $C_a \le C_j \le C_b$ for all $j$, and the approximated visual colors $C'_j$ should be as close as to the original quantized color $C_j$. We thus define a color dissimilarity energy $E_d$ as a function:

$$E_d = \sum_{j=1}^{q} ||C_j - C'_j||, \tag{4}$$

where $||C_j - C'_j||$ denotes the $L^2$ norm (the Euclidean distance) between $C_j$ and $C'_j$ in CIELAB color space. The predicted visual color $C'_j$ is defined similar to the one in Equation 3 and is reformulated as follows:

$$C'_j = \sum_{i=1}^{L} \omega_i \times C_{j,i} \tag{5}$$

where $C_{j,i}$ represents the base color used in the $i^{th}$ layer for synthesizing the color $C_j$. Note that the contribution of canvas color can be removed due to small contribution to the final color. Figure 6(a) shows a sorted color ramp with base colors $C_a$ (yellow) and $C_b$ (orange). By minimizing the color dissimilarity $E_d$ in Equation 4, the combinations of base colors are determined to reproduce the predicted visual colors (see Figure 6(b)(c)). The final synthesized color scribbles are presented in Figure 6(d).

### 5.4. Smoothness among Adjacent Color Segments

While rendering scribbles on each individual color segment $Q_j$, as in the case of Figure 7(b), with the color combinations derived from color reproduction process, it leads to color banding artifacts as shown in Figure 7(c). To reduce the artifact, we manage to have

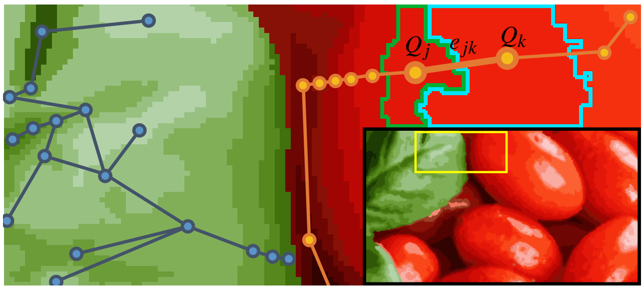



**Figure 8:** A graph is constructed based on the relationship between color segments. Segments $Q_j$ and $Q_k$ (highlighted with sky blue and green boundaries respectively) which are in the same color ramp and have a share boundary are regarded as adjacent segments. The adjacency relationship is denoted by an edge $e_{jk}$ in the graph.

scribbles drawn across the neighboring segments if there are two adjacent color segments with common base color being used. Figure 7(d) shows the effect of a smooth color transition between adjacent color segments. In Figure 7 where $Q_j$ and $Q_{j+1}$ are adjacent segments, then the energy of smoothness $E_s$ can be defined as

$$E_s = \sum_{j=1}^{q-1} \sum_{i=1}^{L} \omega_i \times \|C_{j,i} - C_{j+1,i}\|, \tag{6}$$

where $\omega_i$ is the weighting factor at layer $i$. $C_{j,i}$ and $C_{j+1,i}$ represent the colors of the adjacent segments $Q_j$ and $Q_{j+1}$ at layer $i$, respectively. To simplify the computation, for every adjacent segments, we only consider the color segment pairs at the same layer. The objective function for minimization can then be defined as balancing the two energy functions:

$$E = \alpha \times E_d + (1-\alpha) \times E_s \tag{7}$$

The contribution between color dissimilarity $E_d$ and smoothness $E_s$ is balanced by a weighting factor $\alpha$ where $\alpha \in [0,1]$. Large $\alpha$ tend to ignore the smoothness between adjacent colors while small $\alpha$ tend to sacrifice the color similarity. Based on the experiment, in our implementation, $\alpha$ has been set to 0.4 for deriving some nice results.

## 6. Color Scribble Image Synthesis

To apply the proposed layer-based color dithering model on an input color image, we first quantize the image based on the derived color ramps (see Section 4) and apply the flood-fill algorithm to extract a set of disjointed color segments. For those color segments that are associated with the same color ramp, we run the following process: (i) constructing an adjacency graph among the color segments and computing an optimal assignment of base colors to individual segments; (ii) determining a drawing sequence based on the adjacency relationship to merge the color segments with similar base colors at the same layer. In the following, we shall elaborate the above process in details.

### 6.1. Graph Construction and Optimal Color Assignment

Given a set of quantized color segments, we construct a graph $G$, where each color segment $Q_j$ represents a node and an edge $e_{jk}$ connects two nodes, $Q_j$ and $Q_k$, of two segments share the same boundary (see Figure 8). Our goal in this step is to compute an optimal color assignment for each color segment such that the objective function defined in Equation 7 is minimized. Note that the smoothness term in Equation 6 is revised to adapt to the adjacency graph $G$ as follows:

$$E_s = \sum_{e_{jk} \in G} \sum_{i=1}^{L} \omega_i \times \|C_{j,i} - C_{k,i}\|, \tag{8}$$

Since a target color can be approximated by drawing $L$ layers of color scribbles with two adjacent base colors that enclose the target color (Section 5.3), we regard a sequence of color layers as a label. Figure 6(b) illustrates some example labels with four color layers. Then, the optimization problem can be formulated as a labeling problem where each label corresponds to a predicted visual color represented by a sequence of color layers. We employ a multilabel graph cut algorithm [BVZ01, BV06] to approximate a local minimum.

### 6.2. Drawing Sequence Scheduling

Once the colors, in terms of layers of base colors, has been determined for every color segments, the color scribbles can be drawn from the bottom layer to the top layer in order. However, instead of drawing scribbles inside each color segment independently, different layers of scribbles with common colors could be drawn across segments to smooth out the boundary if they are adjacent.

A greedy method to schedule the drawing sequence is introduced. Based on the proposed dithering model, we have derived the colors of each layers from top to bottom for each color segment. The realization of smooth color transition between adjacent segment is achieved by a scheduling of scribble drawing sequence throughout all the color segments of an image. The sequence is determined by considering the color of the top layer currently being processed in each segment. If two color segments $Q_j$ and $Q_k$ are adjacent (i.e. $e_{jk}$ exists in Figure 8) and the colors being applied to the current top layers of each segment are the same, then the two segments are grouped into one segment and pushed into a stack. Repeating the grouping process until no layer is remained, we then pop out the color and group of segments from the stack to draw the color scribbles in order. We illustrate the scribble drawing sequence by a case of five regions as shown in Figure 9. For drawing sequence illustration of real image, please refer to the supplementary video.

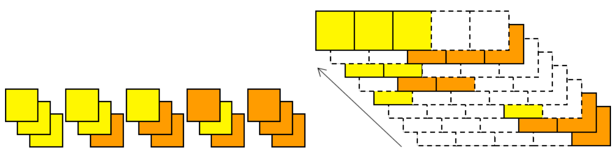
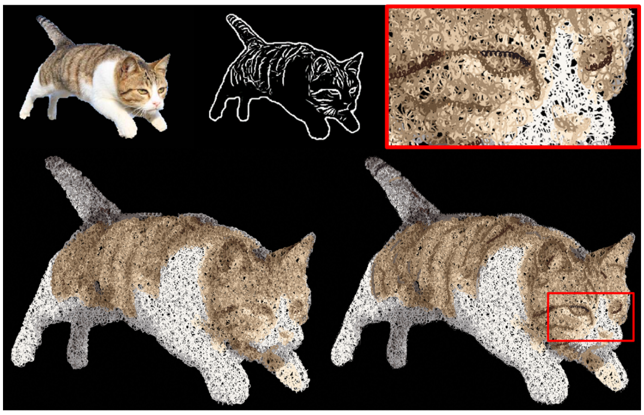



**Figure 9:** Instead of drawing color scribbles sequentially region by region as in (a) which will result in undesired color banding artifact, we manage to group up the adjacent regions and draw scribbles across the boundaries as shown in (b).



without edge scribbles　　　　with edge scribbles

**Figure 10:** Feature enhancement by simply adding scribbles along the provided feature edges.

### 6.3. Enhancement

To better preserve object features from input images, some enhancement methods can be further applied. First, feature map is given by applying modern edge detection operators such as flow-based Difference-of-Gaussians (FDoG) [KLC07] or by manually specifying the feature strokes. For example, if two color segments $Q_j$ and $Q_k$ are adjacent but contain feature pixels to be preserved, then the adjacency relationship (the edge $e_{jk}$) can be removed from the graph to avoid applying smoothness term between two segments. Moreover, to even highlight features of small and thin lines such as the contour of eyes (see Figure 10), we can add an extra scribble layer by directly drawing scribbles along some specific feature lines. The colors of the feature scribbles are chosen from the base colors which minimize the color difference between the colors of pixels defined in the feature map. Figure 10 shows the difference before and after applying the proposed feature enhancement.

## 7. Results and Evaluation

We have tested our method on a wide variety of input images to generate various visually pleasing color scribble images. Some results generated by our system using different stroke patterns can be found in Figure 1 and Figure 12. We refer the readers to the supplementary material for a complete gallery. In general, most of the examples can be accomplished within a minute or two and an 18M-pixel canvas is good enough to generate a delicate image with clear and recognizable color scribble patterns. Table 1 lists the run time performance of our system for generating the color scribble images shown in Figure 1 and Figure 12. We can see that the main bottleneck lies in the rendering of scribble patterns, which took approximately a minute, and the time complexity is proportional to the size of the output canvas. The second bottleneck is the dithering process, which solves the labeling problem, and the time complexity depends on the complexity of the quantized image (i.e., the number of quantized color segments).

In the following, we extensively evaluate the effectiveness of our system by conducting several experimental studies, including: (i) comparision with baseline and state-of-the-art methods; (ii) evaluation of the sensitivity of different parameters settings; and (iii) conducting a user study to justify our methodology.

| Test images | Quantization | Dithering | Scheduling | Rendering |
|---|---|---|---|---|
| *Rose* | 0.289s | 0.350s | 0.080s | 44.7s |
| *Owl* | 0.478s | 0.919s | 0.139s | 33.9s |
| *Rocks* | 0.601s | 1.358s | 0.159s | 51.9s |
| *ClownFish* | 0.904s | 1.346s | 0.241s | 34.7s |
| *Building* | 0.539s | 1.069s | 0.153s | 40.8s |
| *Alien* | 0.531s | 0.540s | 0.114s | 40.0s |
| *Strawberries* | 0.517s | 0.594s | 0.141s | 39.3s |
| *Eagle* | 0.439s | 0.412s | 0.113s | 36.3s |
| *Apples* | 0.420s | 0.442s | 0.103s | 36.2s |
| *Portrait* | 0.837s | 0.479s | 0.123s | 41.0s |
| *Elephant* | 0.459s | 0.237s | 0.070s | 37.5s |
| *Sculpture* | 0.472s | 0.840s | 0.098s | 34.3s |
| *Motorcycle* | 0.518s | 0.697s | 0.187s | 45.1s |
| *Cottage* | 0.586s | 0.732s | 0.154s | 44.7s |

**Table 1:** The execution time (in seconds) for the images shown in Figure 1 and Figure 12.

### 7.1. Comparison with baselines

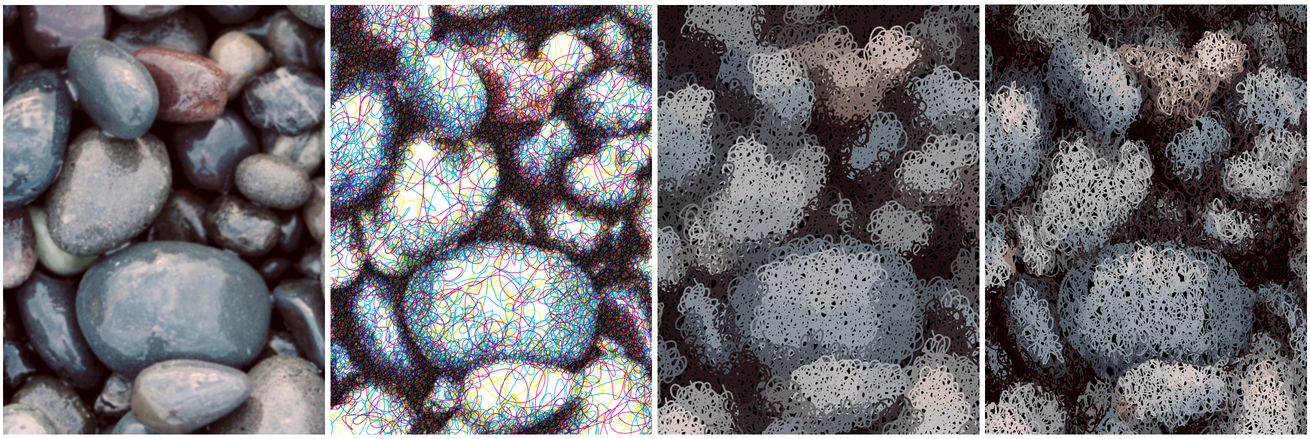
In this section, we compare the visual quality of our results with two baseline methods as follows.

- A naïve extension of [CLLC15], which generates color scribble image by blending individual circular scribble results from different color channels (i.e., CMYK).
- Generating single layer of color scribbles for individual quantized image segments with the corresponding quantized color.

Figure 11 shows a visual example of this experiement. We can easily tell that while the color-version of [CLLC15] can not faithfully reconstruct the color appearance of input image (see Figure 11(b)), the single-layer approach produces only mediocre results using a large number of colors (107 colors in this case) and may still suffer visible boundaries between quantized image segments (see Figure 11(c)). On the contrary, our result uses only 12 colors and utilizes the advantage of layer-based color dithering to faithfully reconstruct the appearance of input image without introducing significant color banding artifacts (see Figure 11(d)).

### 7.2. Comparison with state-of-the-arts

In this section, we compare the quality of our results with exisiting state-of-the-art methods, including (i) a patch-based texture synthesis method [EF01]; (ii) a CNN-based style transfer method [GEB16]; and (iii) a GAN-based image-to-image translation method [IZZE17]. As shown in Figure 13(a), the texture synthesis approach may lead to visible artifacts of fragmented and repeated stroke patterns. The CNN-based style transfer also fails to reproduce the shape and structure of input scribble patterns as shown in Figure 13(b). Moreover, such kind of supervised style transfer reconstructs not only the style but also the color of reference image. Therefore the color of generated image might be inconsistent with original target image. Another interesting comparison will be against the GAN-based method, which is capable of learning and extracting the style among a paired images. We adopted the



(a) (b) (c) (d)

**Figure 11: Comparison with baseline approaches.** (a) Input image. (b) Result generated using a naïve extension of [CLLC15]. (c) Result generated by applying single-layer scribbles to individual image segments using the quantized colors. (d) Our result.

conditional GAN architecture proposed by Isola *et al.* [IZZE17], and trained the model using a dataset with 4000 image pairs. Note that due to the scarce resource of artworks, we used our system to generate color scribble images for preparing the trainning dataset. A visual comparison of some examples can be found in Figure 14. At the first glance, their results seem quite convincing in terms of capturing the scribbling style when comparing to other previous methods. However, when we take a close examination, we can observe some replicated patterns and the generated scribbles are barely traceable (see Figure 14(b)). Moreover, none of the above methods can guarantee generating monochromatic scribbles and control the number of colors in the synthesized results.

### 7.3. Parameter Setting

The results generated by our system can be affected by some scribble parameters such as the stroke configuration of scribble generator and the number of base colors selected. In the following discussions, we will demonstrate how these parameters affect the visual quality of color scribbles by using the circular scribble generator introduced by Chiu et al. [CLLC15].

**Stroke configurations.** Different scribble configurations may result in different scribble coverage ratio. As mentioned in Section 5.2, a low coverage ratio requires higher number of scribble layers to approximate the expected visual color while a high coverage ratio will result in poor color similarity. For example, the stroke configurations in Figure 15(b) and Figure 15(d) will result in high coverage ratio such that the scribble patterns are more difficult to be recognized. Moreover, with high coverage ratio, when multiple layers are applied, the top-most layer will contribute more to the final visual color. That is why Figure 15(b) and Figure 15(d) look more reddish than Figure 15(a) and Figure 15(c) in the bottom row. Therefore, with proper settings of radius size and center velocity, we can have every single layer of scribbles being drawn with proper coverage ratio. In our system, the coverage ratio is very close to 50% by a careful settings of scribble parameters. Figure 15(a) and Figure 15(c) illustrate the results of such scribble parameters settings with coverage ratio close to 50%. We may also derive different visual styles while maintaining the color consistency in the final results. For example, the coverage ratio of the scribble parameters settings in the three cases of Figure 16 are all close to 50%. By applying our dithering model on the same input test image, our system can generate different visual styles but still retain the color similarity to the original input color image.

**Number of base colors.** The number of base colors selected for a color ramp also affects the quality of dithering results. Figure 17 demonstrates cases with the number of base colors from 2 (top row) to 7 (bottom row). The base colors are sampled from the yellow-red color ramp and are used to reproduce the yellow-red color ramp in color scribbles. The color transition from yellow to red becomes smoother when the number of base colors increases.

### 7.4. User Study

We have conducted a preliminary user study to further justify our work in comparison with other approaches. We prepared 23 samples and synthesized four color scribble images for each sample using the following approaches: (1) Image-to-Image Translation (GAN) [IZZE17]; (2) Deep Image Analogy [LYY*17]; (3) our method with three base colors per color ramp; and (4) our method with six base colors per color ramp. The user study is designed to let the participant carefully evaluate how well the synthesized color scribble images resemble the artistic style at both macroscopic and microscopic views. Similar to the pixel-based color dithering, the color presentation should be judged at the macroscopic view. While at the microscopic view, the structures or patterns that render the final color is the key factor to be assessed.

There are 10 participants involved in the study and each of them has to complete 23 trials. During each trial, the four color scribble images are shown in random order and the participant was asked to sort the results subjectively according to the resemblance to a given artwork at macroscopic and microscopic views. The statistics indi-

**Figure 12:** Color scribble images synthesized using our system.

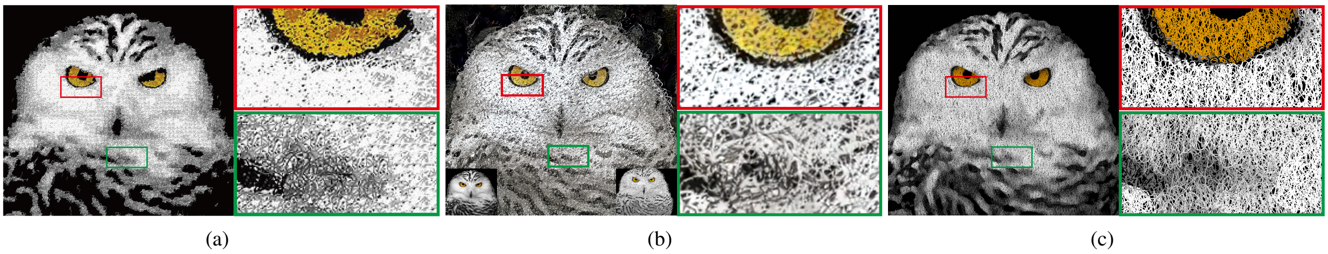


**Figure 13: Comparison with texture synthesis and learning-based style transfer.** (a) The result generated by Image Quilting [EF01] may produce fragmented and repeated texture patterns. (b) The result generated by a CNN-based style transfer [GEB16] may fail to reproduce the shape and structure of input scribble patterns. (c) Our result show superior visual quality when comparing to the other two methods.

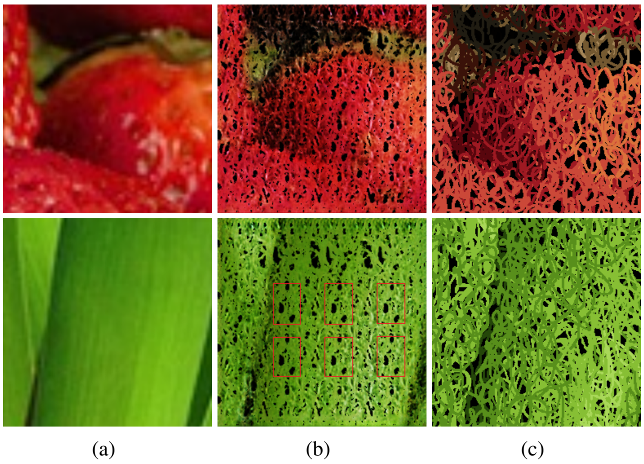


**Figure 14: Comparison with GAN-base style transfer.** (a) Source images. (b) Results generated by conditional GANs [IZZE17]. (c) Our results. Note that our method can generate traceable monochromatic scribble lines without noticeable replicated patterns.

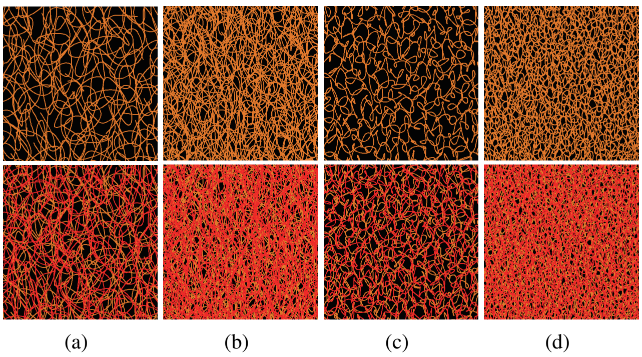


**Figure 15:** Different scribble configurations lead to different visual results. Top row: single layer is applied; Bottom row: two layers are applied. From left to right: (a) default configuration used in our system. (b) larger scribble radius. (c) smaller scribble radius and higher center velocity. (d) smaller scribble radius and lower center velocity.

cate that 80% and 84% of participants report that our approaches best resemble to the artist's color scribble artworks at macroscopic and microscopic views. Moreover, we also found that 80% of participants consider that the color presentation with a higher number of base colors is superior to the one with fewer base colors using our approach at the macroscopic view.

## 8. Conclusion and Discussion

A system to synthesize the color scribble image is introduced. Different from the point-based dithering approaches, we devise a novel layer-based color dithering model to resolve the problems of color reproduction and color banding artifact on the scribble level. The results shows that the proposed method can generate a fine detailed color scribble image with only a few monochromatic colors for scribbles to approximate the original input color image. Besides, the dithering model can well adapt to different scribble styles and the generated results are superior to the state-of-the-art texture synthesis or neural style transfer in the visual quality.

**Limitation.** Although the proposed method can generate delicate color scribble images for most of the input images, there are still limitations. First, artifacts might occur in the low contrast regions due to the quantization error (see the missing dark spots of strawberries in Figure 14(c)). Second, high-frequency features such as furs or thin lines will lead to small and fractional segments where the scribble patterns cannot be rendered effectively (see Figure 18).

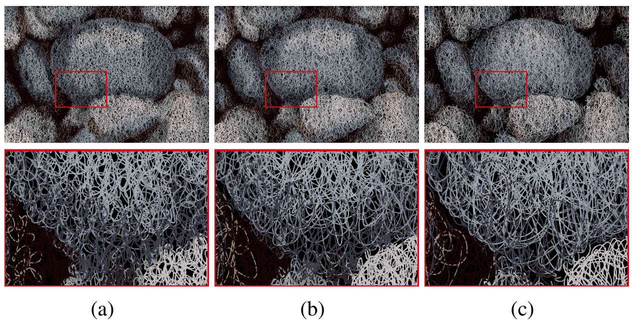


**Figure 16:** From left to right, the center velocity $V_c$, scribble radius $r_{max}$, and sample radius $r_{sample}$, which are the parameters of circular scribble model [CLLC15], are given as $(V_c, r_{max}, r_{sample})$ = $(2,5,5)$, $(2.5,10,6)$, and $(3.5,20,7)$, respectively.

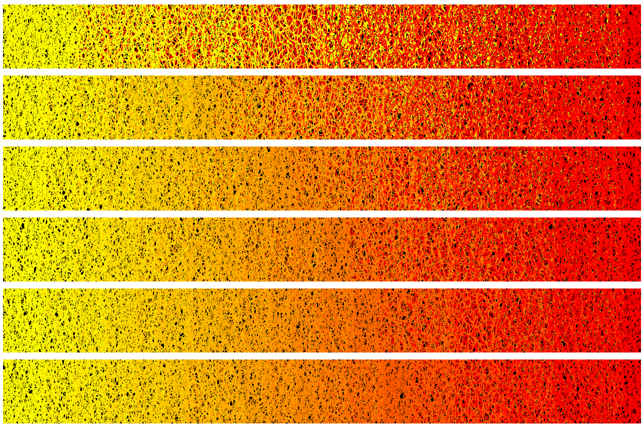


**Figure 17:** From top to bottom: color dithering results by increase the number of base colors from 2 to 7.

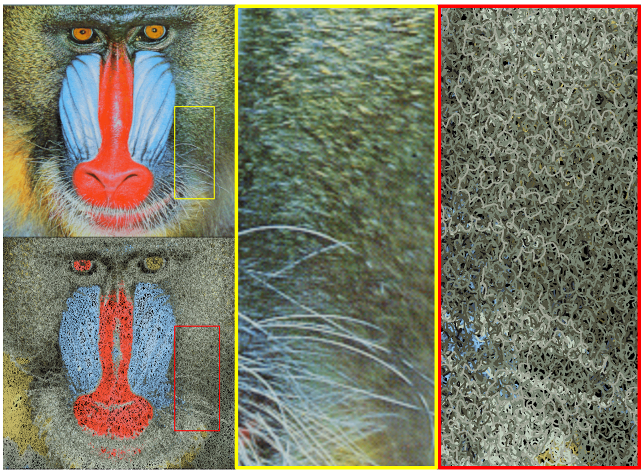


**Figure 18:** An example of feature lost due to small and thin line features.

This artifact can be somewhat alleviated by the features enhancement presented in Section 6.3. Third, our current dithering model is based on the opaque color scribbles, inspired by the digital artwork in Figure 2. Our system does not adapt to the dithering of transparent inks such as cyan, magenta, and yellow, which requires a complex visual color evaluation model for those randomly intersected color scribbles.

**Future work.** There are several interesting directions worthy of further exploration in the future. (i) Instead of using a constant coverage ratio per layer, the skilled scribble artists may use dynamic coverage ratio with the different number of scribble layers to render the final colors. However, making both the coverage ratio and layer number variables will increase the complexity of the model and thus requires further consideration. (ii) We currently using the opaque scribbles for drawing. More interesting interaction and results could be achieved by accounting for the transparency of strokes. To this end, we need to revise the visual color evaluation function with a more complex color blending equation. (iii) The proposed dithering model can be beneficial to digital fabrication, such as 3D printing using additive manufacturing, to simulate the color composition of multiple thin layers.

## References


[AASP17] AKSOY Y., AYDIN T. O., SMOLIĆ A., POLLEFEYS M.: Unmixing-based soft color segmentation for image manipulation. *ACM Transactions on Graphics (TOG) 36*, 2 (2017), 19. 3

[ALMPHS10] ASCENCIO-LOPEZ I., MERUVIA-PASTOR O., HIDALGO-SILVA H.: Adaptive incremental stippling using the poisson-disk distribution. *Journal of Graphics, GPU, and Game Tools 15*, 1 (2010), 29–47. 2

[BH13] BABAEI V., HERSCH R. D.: Juxtaposed color halftoning relying on discrete lines. *IEEE Transactions on Image Processing 22*, 2 (2013), 679–686. 2

[BV06] BOYKOV Y., VEKSLER O.: Graph cuts in vision and graphics: Theories and applications. In *Handbook of mathematical models in computer vision*. Springer, 2006, pp. 79–96. 6

[BVZ01] BOYKOV Y., VEKSLER O., ZABIH R.: Fast approximate energy minimization via graph cuts. *IEEE Transactions on pattern analysis and machine intelligence 23*, 11 (2001), 1222–1239. 6

[CLLC15] CHIU C.-C., LO Y.-H., LEE R.-R., CHU H.-K.: Tone- and feature-aware circular scribble art. *Computer Graphics Forum 34*, 7 (2015), 225–234. 1, 2, 4, 7, 8, 10

[EF01] EFROS A. A., FREEMAN W. T.: Image quilting for texture synthesis and transfer. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques* (2001), ACM, pp. 341–346. 3, 7, 10

[FS76] FLOYD R. W., STEINBERG L.: An adaptive algorithm for spatial greyscale. *Society for Information Display 17*, 2 (1976), 75–77. 2

[GEB16] GATYS L. A., ECKER A. S., BETHGE M.: Image style transfer using convolutional neural networks. In *Computer Vision and Pattern Recognition (CVPR), 2016 IEEE Conference on* (2016), IEEE, pp. 2414–2423. 3, 7, 10

[GP88] GERVAUTZ M., PURGATHOFER W.: A simple method for color quantization: Octree quantization. In *New trends in computer graphics*. Springer, 1988, pp. 219–231. 3

[Hec82] HECKBERT P.: Color image quantization for frame buffer display. *SIGGRAPH Comput. Graph. 16*, 3 (July 1982), 297–307. URL: http://doi.acm.org/10.1145/965145.801294, doi:10.1145/965145.801294. 3

[HJO*01] HERTZMANN A., JACOBS C. E., OLIVER N., CURLESS B., SALESIN D. H.: Image analogies. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques* (2001), SIGGRAPH '01, ACM, pp. 327–340. 3

[HXM*16] HUANG H.-Z., XU K., MARTIN R. R., HUANG F.-Y., HU S.-M.: Efficient, edge-aware, combined color quantization and dithering. *IEEE Transactions on Image Processing 25*, 3 (2016), 1152–1162. 3

[IZZE17] ISOLA P., ZHU J.-Y., ZHOU T., EFROS A. A.: Image-to-image translation with conditional adversarial networks. *CVPR* (2017). 3, 7, 8, 10

[JH05] JANG S., HONG H.-K.: Stippling technique based on color analysis. In *Pacific-Rim Conference on Multimedia* (2005), Springer, pp. 782–793. 2

[KB05] KAPLAN C. S., BOSCH R.: Tsp art. In *Renaissance Banff: Mathematics, Music, Art, Culture* (Southwestern College, Winfield, Kansas, 2005), Sarhangi R., Moody R. V., (Eds.), Canadian Mathematical Society, Bridges Conference, pp. 301–308. 2

[KLC07] KANG H., LEE S., CHUI C. K.: Coherent line drawing. In *Proceedings of the 5th international symposium on Non-photorealistic animation and rendering* (2007), ACM, pp. 43–50. 7

[LAG98] LAU D. L., ARCE G. R., GALLAGHER N. C.: Green-noise digital halftoning. *Proceedings of the IEEE 86*, 12 (1998), 2424–2444. 2

[Llo82] LLOYD S.: Least squares quantization in pcm. *IEEE transactions on information theory 28*, 2 (1982), 129–137. 4

[LYY*17] LIAO J., YAO Y., YUAN L., HUA G., KANG S. B.: Visual attribute transfer through deep image analogy. *ACM Trans. Graph. 36*, 4 (July 2017), 120:1–120:15. URL: http://doi.acm.org/10.1145/3072959.3073683, doi:10.1145/3072959.3073683. 3, 8

[MP92] MITSA T., PARKER K. J.: Digital halftoning technique using a blue-noise mask. *JOSA A 9*, 11 (1992), 1920–1929. 2

[MV02] MESE M., VAIDYANATHAN P.: Recent advances in digital halftoning and inverse halftoning methods. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications 49*, 6 (2002), 790–805. 2

[Neu37] NEUGEBAUER H. E. J.: Die theoretischen grundlagen des mehrfarbenbuchdrucks. *Zeitschrift fur wissenschaftliche Photographie Photophysik und Photochemie 36*, 4 (1937), 73–89. 4

[OH99] OSTROMOUKHOV V., HERSCH R. D.: Multi-color and artistic dithering. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques* (1999), ACM Press/Addison-Wesley Publishing Co., pp. 425–432. 2

[Ost01] OSTROMOUKHOV V.: A simple and efficient error-diffusion algorithm. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques* (2001), ACM, pp. 567–572. 2

[PHK*00] PUZICHA J., HELD M., KETTERER J., BUHMANN J. M., FELLNER D. W.: On spatial quantization of color images. *IEEE Transactions on Image Processing 9*, 4 (2000), 666–682. 3

[PHWF01] PRAUN E., HOPPE H., WEBB M., FINKELSTEIN A.: Real-time hatching. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques* (2001), ACM, p. 581. 2, 3

[PQW*08] PANG W.-M., QU Y., WONG T.-T., COHEN-OR D., HENG P.-A.: Structure-aware halftoning. *ACM Transactions on Graphics (SIGGRAPH 2008 issue) 27*, 3 (2008), 89:1–89:8. 2

[Sec02] SECORD A.: Weighted voronoi stippling. In *Proceedings of the 2nd international symposium on Non-photorealistic animation and rendering* (2002), ACM, pp. 37–43. 2

[Smo13] SMOLKO J. P.: John p. smolko art collections - colored pencil artworks, 2013. URL: http://www.smolkoart.com/. 2

[TFL13] TRESSET P., FOL LEYMARIE F.: Portrait drawing by paul the robot. *Computers & Graphics 37*, 5 (2013), 348–363. 2

[TJT07] TAI Y.-W., JIA J., TANG C.-K.: Soft color segmentation and its applications. *IEEE transactions on pattern analysis and machine intelligence 29*, 9 (2007), 1520–1537. 3

[TLG17] TAN J., LIEN J.-M., GINGOLD Y.: Decomposing images into layers via rgb-space geometry. *ACM Transactions on Graphics (TOG) 36*, 1 (2017), 7. 3

[VF03] VIRMAJOKI O., FRANTI P.: Fast pairwise nearest neighbor based algorithm for multilevel thresholding. *Journal of Electronic Imaging 12*, 4 (2003), 648–660. 2

[WOG06] WINNEMÖLLER H., OLSEN S. C., GOOCH B.: Real-time video abstraction. *ACM Transactions On Graphics (TOG) 25*, 3 (2006), 1221–1226. 3

[WPFH02] WEBB M., PRAUN E., FINKELSTEIN A., HOPPE H.: Fine tone control in hardware hatching. In *Proceedings of the 2nd international symposium on Non-photorealistic animation and rendering* (2002), ACM, pp. 53–ff. 2

[WT11] WONG F. J., TAKAHASHI S.: A graph-based approach to continuous line illustrations with variable levels of detail. *Computer Graphics Forum 30*, 7 (2011), 1931–1939. 2

[WT13] WONG F. J., TAKAHASHI S.: Abstracting images into continuous-line artistic styles. *The Visual Computer 29*, 6-8 (2013), 729–738. 2

[Wu92] WU X.: Color quantization by dynamic programming and principal analysis. *ACM Transactions on Graphics (TOG) 11*, 4 (1992), 348–372. 2

[ZPIE17] ZHU J.-Y., PARK T., ISOLA P., EFROS A. A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. *arXiv preprint arXiv:1703.10593* (2017). 3