DOI: 10.1111/cgf.12761
Pacific Graphics 2015
N. J. Mitra, J. Stam, and K. Xu
(Guest Editors)

*Volume 34 (2015), Number 7*

# Tone- and Feature-Aware Circular Scribble Art

Chun-Chia Chiu† Yi-Hsiang Lo† Ruen-Rone Lee Hung-Kuo Chu

National Tsing Hua University, Taiwan

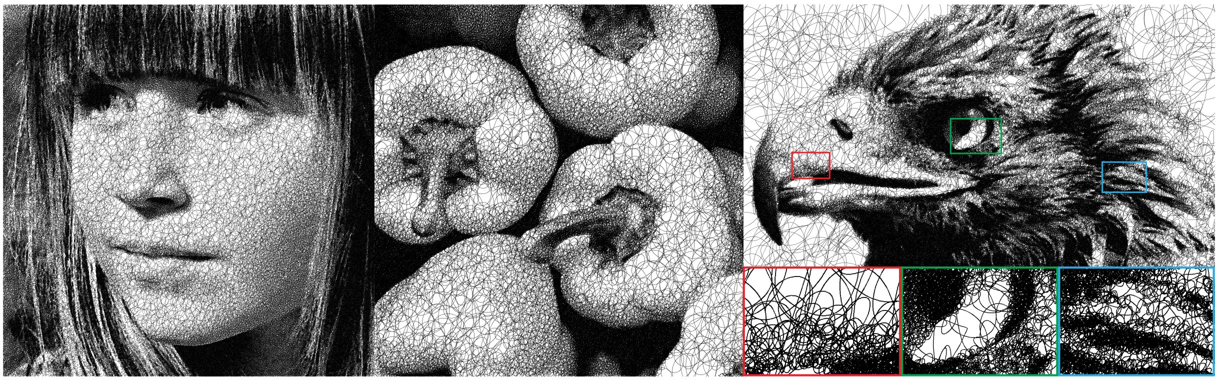


**Figure 1:** *Circular scribble images generated using our approach. Please zoom in properly to see the detailed circular patterns.*

**Abstract**

*Circular scribble art is a kind of line drawing where the seemingly random, noisy and shapeless circular scribbles at microscopic scale constitute astonishing grayscale images at macroscopic scale. Such a delicate skill has rendered the creation of circular scribble art a tedious and time-consuming task even for gifted artists. In this work, we present a novel method for automatic synthesis of circular scribble art. The synthesis problem is modeled as tracing along a virtual path using a parametric circular curve. To reproduce the tone and important edge structure of input grayscale images, the system adaptively adjusts the density and structure of virtual path, and dynamically controls the size, drawing speed and orientation of parametric circular curve during the synthesis. We demonstrate the potential of our system using several circular scribble images synthesized from a wide variety of grayscale images. A preliminary experimental studying is conducted to qualitatively and quantitatively evaluate our method. Results report that our method is efficient and generates convincing results comparable to artistic artworks.*

## 1. Introduction

Scribble art is a kind of illustrative drawing, where the artists use seemingly random and continuous line drawing to depict images or conceptual designs. Unlike the conventional line drawing such as sketching and hatching that use short and straight line segments, scribble artists aim at managing long and continuous curves. In this work, we study a commonly seen scribble pattern, namely circular scribble, where the circular lines are drawn in either clockwise or counter-clockwise direction with varying radius. In circular scribble art, the artistic skills usually show delicate controls over tracing along a virtual path while constantly changing the radius and orientation of circular scribbles to depict a subject of interest as shown in Figure 2.

As an art form, circular scribble art shares a common goal with other stylized line drawing to produce a faithful abstraction of target images in terms of tone reproduction and structure preservation. However, in contrast to conventional line drawing where the stroke patterns can be controlled precisely, circular scribbles generally appear in a form where

† These authors contributed equally to this work.

the continuous loops are everywhere with massive overlaps and intersections (see Figure 2(right)). Therefore, using such irregular shapes to represent tone and structure requires delicate artistic skills thus making the creation of circular scribble art a tedious and time-consuming task.

In this work, we present a novel system to automatically synthesize visually appealing circular scribble art from arbitrary grayscale images. To reproduce the tone and important edge structure in a grayscale image, the system works in two phases. First, the system generates a virtual tracing path that is aware of the tone and edge structure in input image. Specifically, we sample a set of seed points according to the grayscale distribution of input image, and divide them into groups based on color segmentation. The virtual path is created by connecting local paths derived from the linkage of sample points within each group. In this way, we obtain a virtual path with minimal occurrences of crossing edge problem. Then, a parametric circular curve is proposed to draw along the virtual tracing path while the system adaptively adjusts the curve parameters to adapt to a learned tone mapping and respect local edge structure during the synthesis process. As a result, our system generates tone- and structure-preserved circular scribble images that resemble the artworks (see Figure 1). A preliminary experimental result shows that our system is effective and outperforms the state-of-the-art method using texture synthesis.

Our major contributions include:

- An automatic algorithm to synthesize tone- and structure-aware circular scribble art from grayscale images.
- Proposing a virtual path tracing mechanism that accounts for tone distribution as well as avoiding the occurrence of crossing edge structure.
- A novel parametric representation for the circular scribble pattern, which is used to synthesize circular scribbles along the virtual path.
- A circular scribble generator that provides dynamic parameters control over the radius, drawing speed and tilt angle of tracing scribbles based on a learned tone mapping and local edge structure.

## 2. Related Works

Our work is closely related to the synthesis of digital art. Given a 2D image, the synthesis work is trying to analyze and extract the necessary information, and then converting the image pixels into other forms of representation such as varying sized dots, texture patches, line drawings, and so on. Based on the representation style, we classify these related works into point-based, texture patch, and line drawing.

**Point-based Representation.** Halftoning [PQW*08] and stippling [ALMPHS10] are common techniques to represent

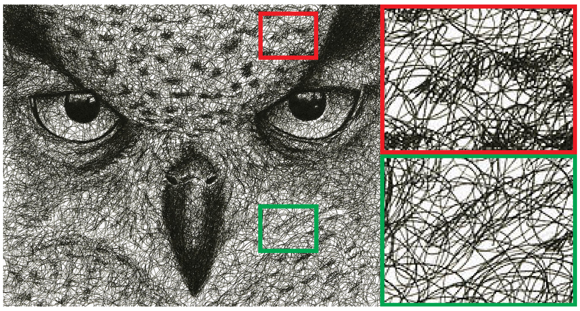


**Figure 2:** *Artist created circular scribble art (©Nathan Shegrud).*

the original image through a set of sampled points. The sample points can vary in size and the distribution must convey as much as to the tone and features of the original input image. Despite its tone and structure similarity to the input image, the output are discrete points without any information to indicate the possible artistic creation process. Our work is different from their works in the final representation. A continuous line drawing together with a drawing sequence to emulate the creation process is generated.

**Texture Patch Synthesis.** Texture synthesis is another approach to convert an input image into a specific stylization. Through a learning process to extract patterns as texture patches from existing artworks, the stylization is synthesized by overlapping and stitching the extracted texture patches [EF01, HJO*01, PHWF01]. Although texture patch synthesis has been proved to be a success in synthesizing some artworks in painting, but it has some limitations especially in duplicating the line drawing style. For short stroke such as hatching line, the texture patches can be used to simulate the tone through the density of strokes. The saliency and shape can also be simulated by small straight lines or by curves. However, for long continuous line drawing, this approach will result in fragmented strokes due to the limited number of texture patterns and the size of texture patches. Therefore, it is difficult to synthesize a nice continuous line drawing artwork.

**Line Drawing.** There are different kinds of line drawing approaches. For example, through well defined filters, the lines can be detected and then connected or converted to some kinds of line drawing arts as can be seen in the works [KLC07, LXJ12, WPFH02]. Berger *et al.* [BSM*13] used a data-driven approach to synthesize a portrait sketching that are close to some specific artistic style collected in the database. But, it just replaces the strokes by the strokes of a specific artist. Robot the Paul [TFL13] is a system that synthesizes a user sketch incrementally through a feedback mechanism by comparing to the original image. Despite the result is very convincing, the line drawing sequence is unpre-

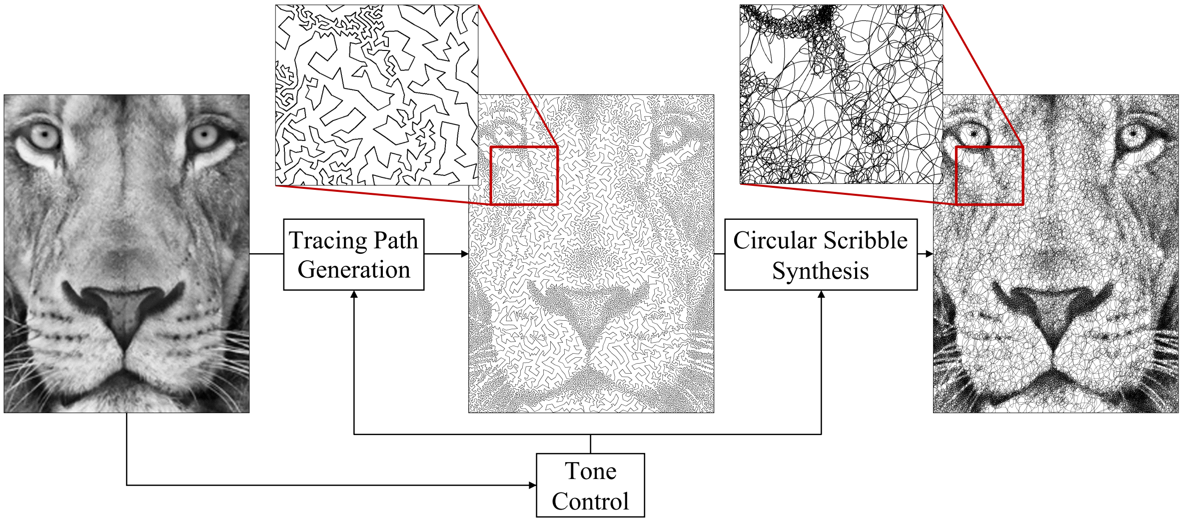



**Figure 3:** *Our system flow consists of tracing path generation and circular scribble synthesis with parameters adjusted by tone control.*

dictable and discontinuous due to the feedbacks are applied to disjoint regions.

Another line of work aims at generating a drawing sequence of continuous lines to depict the outlines and features of input images [BH04, KB05, WT11, WT13a, WT13b]. Our tracing path generation is mainly inspired by the works of [KB05] and [WT13a] where a long continuous circuit is traced over the entire image with varying line space to capture the intensity variation. Kaplan and Bosch [KB05] modeled the path tracing problem as the Traveling Salesman Problem (TSP) and presented TSP Art. In their approach, an intensity-aware sampling is first used to generate sampling points (i.e., cities) on the input grayscale image followed by a heuristic approach to derive connected edges among cities. The resultant path is guaranteed to have no intersection and approximates the luminance of image when looking from a distance. Wong and Takahashi [WT13a] presented continuous-line artistic styles. With the aid from users to provide image segmentation and prescribe directions of tracing path, their system is able to efficiently generate quality results that are superior than TSP Art. However, none of above methods address the problem of preserving feature edges during the path tracing, and thus can not directly apply to our problem without a further consideration.

## 3. System Overview

Our system is consisting of two main stages, a tracing path generation and a circular scribble synthesis, as shown in Figure 3. A continuous tracing path which minimizes feature crossing and self-intersection is first generated. Then, the circular scribbles are synthesized with parameters controlling the scribble size, speed, and orientation, where the circular scribble center is tracing along the given tracing path. A tone control is further introduced to assist the sampling process of tracing path generation as well the parameter adjustment during circular scribble synthesis.

## 4. Tracing Path Generation

Conventional circular scribble art is created by drawing the circular scribble with center following a virtual path. In order to derive a continuous circular scribble art, a virtual tracing path for guiding the center of circular scribble is required. A proper tracing path should have two properties: (1) The path should cover the whole subject which is to be drawn; and (2) it should preserve both structures and details of the subject. For this purpose, we propose a novel tracing path generation algorithm that combines the advantages of both [KB05] and [WT13a], while accounting for the edge preservation as elaborated below.

### 4.1. Sampling

In order to derive a tracing path which covers the image, a sampling process is required to define the points where the path will pass through. The sampling problem in Euclidean space is similar to the cities distribution in TSP problem, where the density of sample points should well capture the local intensity of the source image. Hence, we employ the adaptive sampling strategy proposed by Ascencio-Lopez *et al.* [ALMPHS10], which controls the sampling density based on image intensity and presents desired blue noise property. Note that while there are other possible alternatives [Sec02, KCODL06] in the sampling process, we found [ALMPHS10] provides a better trade off between quality and computational complexity. In our implementation, we first scale the longest dimension of input image to 4000 pixels, and set the minimum and maximum sampling radius respectively to 10 pixels and 45 pixels to generate sampling points.

### 4.2. Path Construction

After the cities are distributed as sample points, we should then have these points be linked together as a continuous path. The path construction of TSP can be solved by applying the K-L heuristic approximation algorithm [KL70] with time complexity $O(n^{2.2})$. However, the execution time grows exponentially when the number of sample points becomes large. Thus, similar to the algorithm presented in [WT13a], we design a path construction algorithm that executes in three steps, which are superpixel segmentation, local tracing path construction, and merging.

**SLIC Superpixel Segmentation.** Similar to the modern tiling algorithms in image processing, we first segment the source image into disjoint parts in order to parallelize the path construction algorithm. Instead of using a grid-like shape, we generate an image segmentation where each image segment would respect both the color distribution and edge structure of original image. Specifically, we adopt the state-of-the-art image over-segmentation, called SLIC, which adapts a k-means clustering approach to efficiently generate superpixels [ASS*12]. SLIC offers intuitive parameters control over the amount of superpixels and their compactness. We use default settings of the number of superpixels (200) and the compactness (20) to generate all results. Since the boundaries of superpixels adhere well to the edges of image, such property would reduce the possibility of generating paths that cross the edge features, and thus result in better quality in the synthesized circular scribbles. We will elaborate the feature preservation in Section 5.2.

**Local Tracing Path Construction.** After the image segmentation, we apply the path construction method of K-L algorithm to the sample points covered by a superpixel segment. Without data dependency, the associated paths for each superpixel are generated concurrently to speed up the local path construction process.

**Merging.** The tracing paths of each superpixel are all close circuits. We utilize the property of Hamiltonian circuit to perform this merging step. The merging process can be illustrated by Figure 4. First, we define that any two circuits are adjacent if their corresponding superpixels are adjacent in the original image. For any two adjacent circuits as in Figure 4(a), an edge is removed in each small circuit (see Figure 4(b)). Then, the edge pairs $(\overline{A_iB_{j+1}}, \overline{A_{i+1}B_j})$ or $(\overline{A_iB_j}, \overline{A_{i+1}B_{j+1}})$ which connect the two opened paths with shorter length are evaluated (see Figure 4(c)). The merged circuit is finally derived after connecting the shorter edges

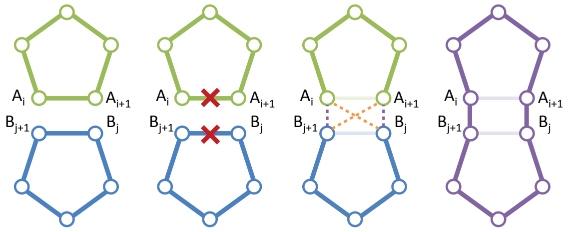



**Figure 4:** *(a) Two adjacent circuits (b) An edge is removed (c) Shorter edge pair is chosen (d) The merged circuit*

(see Figure 4(d)). For every two adjacent circuits, we consider the energy function:

$$E_{merge}(A, B) = \min_{\substack{e_i^A \in A \\ e_j^B \in B}} \{E_c(e_i^A, e_j^B) - E_r(e_i^A, e_j^B)\} \quad (1)$$

where

$$E_c(e_i^A, e_j^B) = \min\{|A_i - B_j| + |A_{i+1} - B_{j+1}|, \\ |A_i - B_{j+1}| + |A_{i+1} - B_j|\} \quad (2)$$

and

$$E_r(e_i^A, e_j^B) = |A_i - A_{i+1}| + |B_j - B_{j+1}| \quad (3)$$

The edge $e_i^A$ denotes the edge $\overline{A_iA_{i+1}}$ which starts from vertex $A_i$ and ends at vertex $A_{i+1}$ in circuit $A$. Likewise, $e_j^B$ denotes the edge $\overline{B_jB_{j+1}}$ in circuit $B$. We traverse all possible cases to find out the minimum energy of merging a pair of circuits. With this bottom-up and pair-by-pair merging step, we can finally obtain an Eulerian tracing path.

**Evaluation.** Although the K-L algorithm will result in an approximately optimal TSP circuit without crossover lines, our method tend to accept some crossover lines but reduce the chance to generate lines which might cross the image feature edges. The final output path is regarded as a virtual tracing path for our circular scribble generator to follow along. Figure 5 illustrates a close-up look of the tracing paths with and without superpixel segmentation. As highlighted in the red boxes of Figure 5, the generated circular scribbles in Figure 5(b) did not cross the image features as compare to the result shown in Figure 5(a). Even though our method does contains small crossover lines in the final tracing path, the generated circular scribble art makes only a few artifacts which are almost unaware of.

## 5. Circular Scribble Synthesis

A circular scribble model is first introduced to define how the scribble is generated with respect to some control parameters. Besides, in order to better preserve the salient features, dynamic adjustment to the scribble size with respect to the salient edge map during scribble synthesis is also proposed.

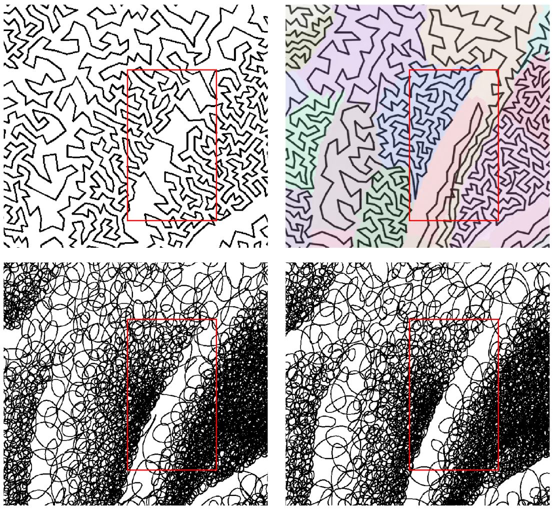



**Figure 5:** *(a) Top: tracing path without superpixel segmentation; Bottom: circular scribble art without superpixel segmentation. (b) Top: tracing path with superpixel segmentation; Bottom: circular scribble art with superpixel segmentation provides better feature preservation in appearance.*

### 5.1. Scribble Model

The circular scribble is very similar to a trochoid which is a curve defined by the locus of a fixed point on a circle as it rolls along a fixed line. By observing how the circular scribble is drawn by an artist, we extend the concept of rolling circle to a rolling 3D disk as follows:

$$C\text{: } (p,\, r,\, \omega,\, v_c,\, \vec{n}) \rightarrow S \tag{4}$$

where $C$ is the circular scribble generator. $p$ is the center position of the disk moving along a tracing path. $r$ represents the disk radius, which controls the size of circular scribble. Both $\omega$ and $v_c$ are angular velocity and the instantaneous velocity of the moving disk. The vector $\vec{n}$ is the disk normal. By varying these parameters, the circular scribble $S$ is generated as a projection of the fixed point on the rolling disk to the 2D Euclidean space with its center follow through the given tracing path. Figure 6(a) gives an example of circular scribble with $p$ following a straight line and with the remaining parameters being set to some constants. We have derived a tracing path for the disk center to follow in previous section. We will now discuss how to adjust the remaining control parameters of the circular scribble generator such that a circular scribble art can be synthesized with respect to a given image.

**Varying Radius.** Disk radius is changed dynamically while the disk is rolling with its center moving along the tracing path. We adjust the radius with respect to the luminance value $L(p)$ at position $p$ in the given image, where $p$ is a sample on the tracing path and the disk center is currently on $p$. The control of the disk radius can be described as follows:

$$r_{ori}(p) = R_{min} + L(p) \times (R_{max} - R_{min}) \tag{5}$$

where $R_{min}$ and $R_{max}$ are the smallest and largest radius of the circular scribble defined in the scribble generator. Since the luminance value $L(p)$ is in the range of $[0,1]$, thus the value of disk radius will be confined in the range of $[R_{min}, R_{max}]$. Figure 6(b) shows how the circular scribble is drawn with varying radius $r$.

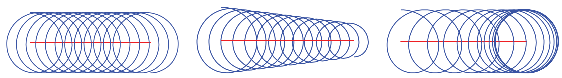



**Figure 6:** *Circular scribbles are drawn from left to right: (a) Original scribble model. (b) Scribble with radius varying from r to* $0.5r$. *(c) Scribble with decreasing center velocity.*

**Varying Angular and Center Velocity.** Similar to the trochoid curve, the ratio $\lambda = \frac{v_c}{\omega}$ dominates the presentation of circular scribble as can be seen in Figure 7. In this paper, in order to generate *circular* scribble, we consistently keep $\lambda < 1$ during the synthesis of a circular scribble art. To control the ratio $\lambda$, we carefully adjust the center velocity $v_c$ and keep the angular velocity $\omega$ constant in practice:

$$V_c(p) = V_{min} + L(p) \times (V_{max} - V_{min}) \tag{6}$$

Equation 6 shows the determination of center velocity $V_c$ with respect to the luminance value $L(p)$ at position $p$ and the range in $[V_{min}, V_{max}]$. Also, Figure 6(c) shows an example of circular scribble with varying center velocity $v_c$.

**Varying Disk Orientation.** To further imitate artistic style of circular scribble, we found that artists tend to tilt the pen in order to draw scribble in certain orientation. In our circular scribble model, we have defined normal vector $\vec{n}$ to cover

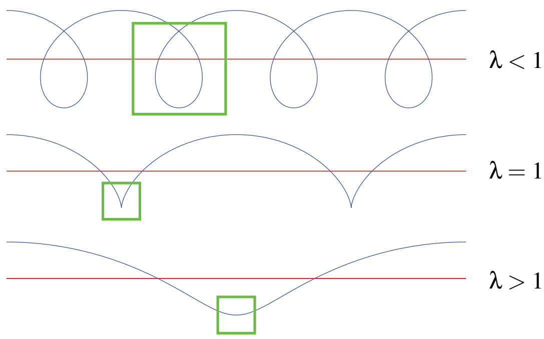



**Figure 7:** *Scribble representations with respect to different ratios of* $\lambda = \frac{v_c}{\omega}$.

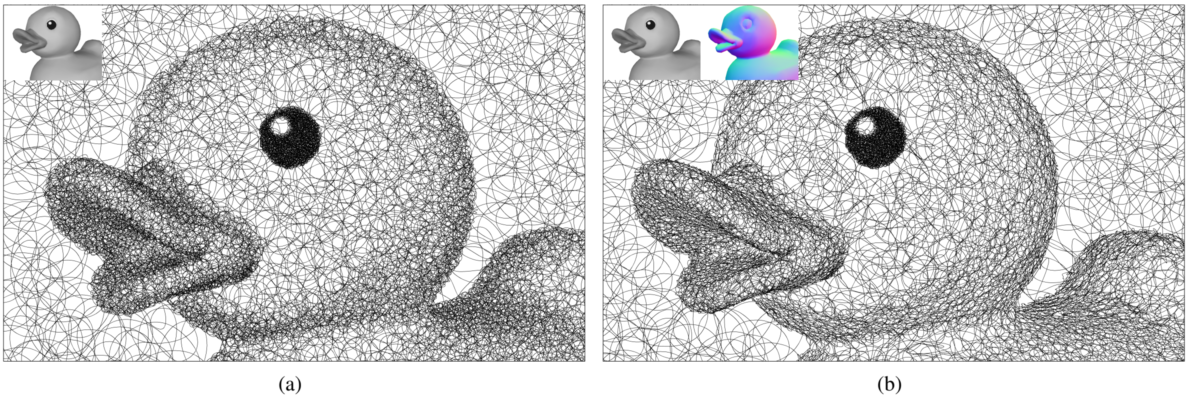



**Figure 9:** *Circular scribble art (a) with orientation unaltered; and (b) orientation altered by using the normal map of 3D model.*

this capability, where $\vec{n}$ is represented by an azimuth angle $\theta$ and a polar angle $\varphi$ in a spherical coordinate system. The normal vector $\vec{n}$ is used to control the shape of circular scribble. Azimuth angle $\theta$ masters the degree of tilt angle, while the polar angle $\varphi$ control the width of a circular scribble. Figure 8(a) and Figure 8(b) demonstrate the resulting circular scribbles by varying the azimuth angle and the polar angle, respectively. Small random jittering to the normal vector can be applied to prevent from generating observable regular scribble patterns. Moreover, an orientation map, such as a normal map of a 3D model, can also be adopted to adjust the orientations during the circular scribble synthesis of a 3D model. Figure 9 demonstrates the results with and without the aid of normal map.

### 5.2. Feature Preservation

The disk radius $r$ has been defined according to the luminance of input image in circular scribble model. However, when the circular scribble is close to the image features

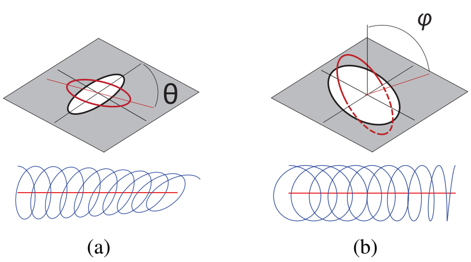



**Figure 8:** *Circular scribbles are drawn from left to right: The normal vector $\vec{n}(\theta,\varphi)$ dominates the shape of scribble. (a) Scribble with azimuth angle $\theta$ varying from $0°$ to -$45°$. (b) Scribble with polar angle $\varphi$ varying from $0°$ to $90°$.*

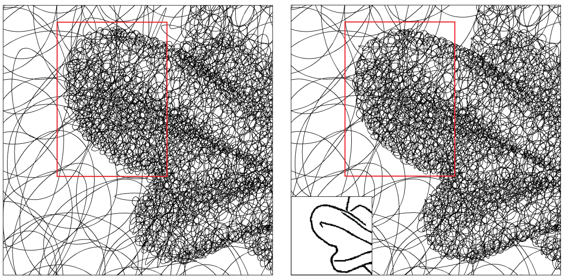
where the radius is larger than the distance between the center and the image feature, the circular scribble might cross the feature boundary which results in overdrawn outside the features as can be seen in Figure 10(a). In order to better preserve the image features, a feature preservation constraint is designed to dynamically adjust the disk radius when the scribble is close to the feature boundary. An edge map, such as the edges derived from FDoG [KD09], can be used to serve as a feature map. Then, the disk radius is properly adjusted by the following rules:

$$r = \begin{cases} R_{min} & \text{if } dist(p,E) < R_{min} \\ dist(p,E) & \text{if } R_{min} \le dist(p,E) \le r_{ori} \\ r_{ori} & \text{if } dist(p,E) > r_{ori} \end{cases} \tag{7}$$

The distance from disk center $p$ to the closest image feature edge in the given edge map $E$ is calculated as $dist(p,E)$. When the distance is smaller than the smallest radius $R_{min}$, then the radius is set the $R_{min}$. If the distance is larger than the current scribble radius, then there is no need to adjust the radius. Otherwise, the radius is adjusted to be the distance between the scribble center to the feature edge. The improved result with feature preservation radius adjustment is shown in Figure 10(b).

## 6. Tone Control

Tone control is essential in deriving a visual pleasing circular scribble art. It is applied on the input image to remap the original luminance values into associated new luminance levels which the subsequence process can refer to. The remapping function is derived as follows. Given a grayscale bar as input image, we first generate a circular scribble result as in the middle of Figure 11. Then, both grayscale bar and the circular scribble result are blurred to simulate the visual luminance from a distance. We then divide the grayscale bar with intensity range within $[0,1]$ into, for example, 256 levels. For each level with luminance intensity $x$ in blurred



(a) (b)

**Figure 10:** *(a) Circular scribble art before feature preservation. (b) After feature preservation, the features are well preserved.*

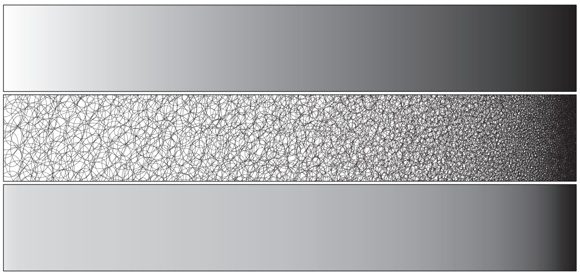
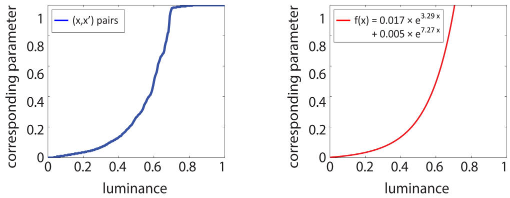


**Figure 11:** *From top to bottom: input grayscale bar, circular scribble result, and blurred circular scribble result.*

grayscale bar, we search for the most similar luminance $x'$ in the blurred circular scribble result, such that $|x - x'|$ is as small as possible. Once we have the output luminance $x'$, the corresponding input luminance $y$ is then retrieved in the grayscale bar. After each $(x, y)$ pair is found, we define a tone control mapping function $y = f(x)$ shown as the blue curve in Figure 13(a). The tone control mapping function $f(x)$ is applied for each sample point in luminance map $L$. That is, for every parameter that needs the luminance value at position $p$ of $L$, we consider $f(L(p))$ instead of directly using the original luminance value $L(p)$. Figure 12 shows the circular scribble results of using original luminance map

(a) (b) (c)

**Figure 12:** *Circular scribble arts (a) without tone control and (c) with tone control. The original input image is shown in (b).*




(a) (b)

**Figure 13:** *Mapping function for tone control. X-axis is the intensity of grayscale bar input. Y-axis is the corresponding intensity to generate an output with better tone similarity. (a) is the plot of 256 sampled pairs with luminance ranging from 0 to 1 and (b) is the mapping function using curve fitting.*

(Figure 12(a)) and tone-controlled luminance map (Figure 12(c)). There are two observations as follows:

**Exponential Mapping.** The tone control mapping function mentioned above is a discrete lookup table. We can apply curve fitting on each $(x, y)$ pair as sample event. The result comes up with a combination of two exponential functions shown as the red curve in Figure 13(b).

**Poor Similarity at High Luminance.** While dealing with high intensity of luminance, our system seems to result in lower intensity for circular scribble. The main reason is that leaving blank area without any scribble passing by will lead to some unpleasant artifacts by the style of circular scribble. Hence, we have decided to draw at least one stroke in every local region, even the region is in high luminance intensity. The quality of tone similarity will be discussed in Section 7.

## 7. Results and Discussion

We demonstrate the robustness of our circular scribble generator by generating visual pleasing results from a wide variety of input images (see Figure 14). Moreover, our circular scribble generator also adapts to generate color circular scribble results as well (see Figure 16). The color circular scribble is a blending of individual circular scribble results from different color channels such as CMYK or RGB. Besides the static image presentation, an animation to imitate the real drawing sequence of a continuous circular scribble can also be generated. By recording the tracing process of our circular scribble, we can generate an animation which resembles a real scribble art drawing. Please refer to the supplementary materials for more results of circular scribble images and animation.

**Quality Assessment and Comparison.** Our scribble generator is a novel style of line drawing. To evaluate the quality of tone and structure similarity, and circular scribble properties, we compare our results with texture transfer [EF01]. Similarity can be quantized by analyzing the well-known quantitative metrics, PSNR and SSIM. Since our generator

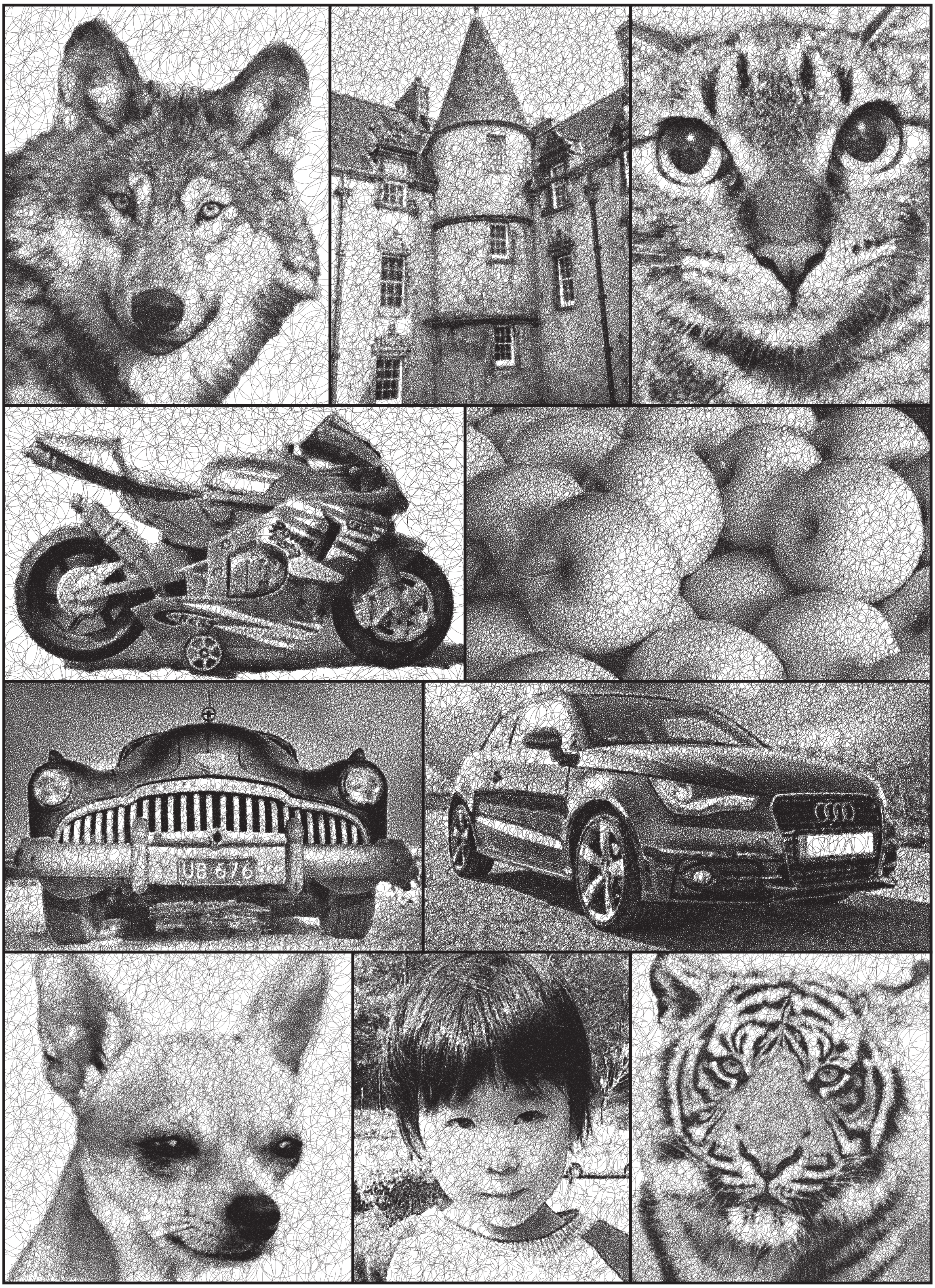
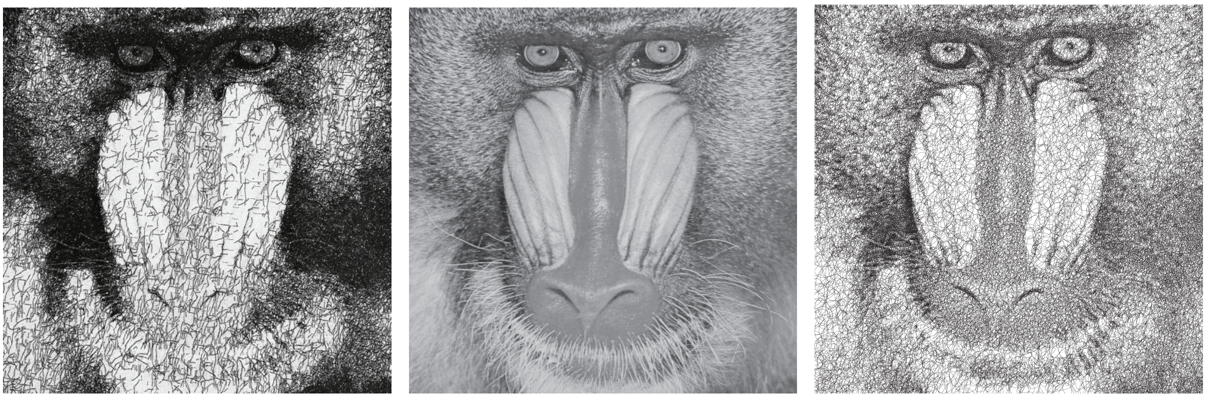



**Figure 14:** *Our generator can adapt to various input image categories. Please zoom in properly to see the detailed circular patterns.*



(a) (b) (c)

**Figure 15:** *Results of (a) patch-based texture transfer and (c) our tone- and feature-aware circular scribble art. The original input image is shown in (b). Please zoom in for more structure details in digital paper version.*

| | Texture Transfer | | Our Results | |
|---|---|---|---|---|
| | PSNR | SSIM | PSNR | SSIM |
| Lion(Fig. 3) | 16.3 | 0.753 | 24.9 | 0.916 |
| Rocks(Fig. 11) | 15.5 | 0.629 | 20.8 | 0.943 |
| Apples(Fig. 14) | 14.3 | 0.638 | 17.6 | 0.897 |
| Baboon(Fig. 15(c)) | 17.8 | 0.873 | 20.4 | 0.958 |

**Table 1:** *Comparing the quality of tone and structure similarity with texture transfer [EF01].*

produces results in high resolution canvas, we scale the circular scribble results down to the original input image size. Next, we apply Gaussian blur to both texture transfer results and scribble art results to derive the corresponding PSNR and SSIM values. Table 1 shows the results with respect to different input images. It is apparent that our results are superior than texture transfer both in tone and structure similarity. Another visual quality assessment is to retain the characteristic of circular pattern and the scribble continuity. They are difficult to be quantized but can be easily compared visually. Figure 15 shows the advantages of our circular scribble generator which preserves both circular pattern and scribble continuity.

**Performance.** All the experimental results are generated on a moderate desktop PC with Intel i5 CPU (3.4 GHz). Table 2 details the running times of our system for generating results shown in this paper and an average of 50MB is reported in system memory consumption. We further compare the timing complexity of tracing path generation with and without using image segmentation. The results indicate that the proposed algorithm (i.e., column "path tracing") is an order of magnitude faster than the one without using image segmentation (i.e., column "direct path tracing"). Overall our system is efficient and can generate circular scribble images within a reasonable amount of time.

| | Sampling | Path Tracing | Scribble Synthesis | Total Timing | Direct Path Tracing |
|---|---|---|---|---|---|
| Lion(Fig. 3) | 7.7s | 13.9s | 12.1s | 33.6s | 4.6m |
| Rocks(Fig. 11) | 7.3s | 15.9s | 12.1s | 35.3s | 11.0m |
| Apples(Fig. 14) | 9.1s | 15.8s | 11.4s | 36.3s | 14.0m |
| Baboon(Fig. 15(c)) | 10.2s | 20.7s | 17.0s | 47.9s | 9.8m |

**Table 2:** *Timing statistics of our circular scribble generator.*

## 8. Conclusions

A novel circular scribble generator is introduced to synthesize a delicate circular scribble art from an input image. The results are convincing with well tone and feature preserved. In this work, we have successfully proposed a systematic way in simulating the process of creating a circular scribble artwork. Through an efficient tracing path construction and a flexible circular scribble generator, with the aid of tone and

**Figure 16:** *A color circular scribble result generated by a naive extension of our system. (©Toni Lankerd)*

feature preserving control, an artwork with a single continuous circular scribble can then be generated digitally.

**Limitations and Future Works.** Despite the delicacy of our approach, there are still some limitations to our circular scribble. First, it is difficult to synthesize the bright tone well due to some minimal scribbles will cover the region even it is completely bright. We have tried to leave blank for those bright region but lead to some unpleasant artifacts to the boundary near the tone transition. Second, the quality of our results rely on the quality of input images. An input with poor contrast and weak feature edges will results in dull output. As for the future works, we expect to further simulate the real drawing with different pen styles, paper materials, and other color circular scribble presentations. Our system also has potential to be integrated into commercial image processing tools such as Adobe® Photoshop.

## Acknowledgements

We are grateful to the anonymous reviewers for their comments and suggestions. The project was supported in part by the Ministry of Science and Technology of Taiwan (102-2221-E-007-055-MY3, 103-2221-E-007-065-MY3, 103-2220-E-007-012 and 104-2220-E-007-016).

## References

[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, 3

[ASS*12] Achanta R., Shaji A., Smith K., Lucchi A., Fua P., Sãijsstrunk S.: SLIC superpixels compared to state-of-the-art superpixel methods. *IEEE Trans. Pattern Anal. Mach. Intell. 34*, 11 (2012), 2274–2282. 4

[BH04] Bosch R., Herman A.: Continuous line drawings via the traveling salesman problem. *Operations research letters 32*, 4 (2004), 302–303. 3

[BSM*13] Berger I., Shamir A., Mahler M., Carter E., Hodgins J.: Style and abstraction in portrait sketching. *ACM Transactions on Graphics (TOG) 32*, 4 (2013), 55. 2

[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. 2, 7, 9

[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, pp. 327–340. 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. 3

[KCODL06] Kopf J., Cohen-Or D., Deussen O., Lischinski D.: Recursive wang tiles for real-time blue noise. *ACM Transactions on Graphics (TOG) 25*, 3 (2006), 509–518. 3

[KD09] Kyprianidis J., Döllner J.: Real-time image abstraction by directed filtering. *ShaderX7-Advanced Rendering Techniques. Charles River Media* (2009), 285–302. 6

[KL70] Kernighan B. W., Lin S.: An efficient heuristic procedure for partitioning graphs. *Bell system technical journal 49*, 2 (1970), 291–307. 4

[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. 2

[LXJ12] Lu C., Xu L., Jia J.: Combining sketch and tone for pencil drawing production. In *Proceedings of the Symposium on Non-Photorealistic Animation and Rendering* (2012), Eurographics Association, pp. 65–73. 2

[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

[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. 3

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

[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. 3

[WT13a] Wong F. J., Takahashi S.: Abstracting images into continuous-line artistic styles. *The Visual Computer 29*, 6-8 (2013), 729–738. 3, 4

[WT13b] Wong F. J., Takahashi S.: Hierarchical design of continuous line illustrations. In *GRAPP/IVAPP* (2013), INSTICC, pp. 131–138. 3

[Yat12] Yates R. C.: *A Handbook On Curves And Their Properties*. Literary Licensing, LLC, 7 2012.