International Journal of Signal Processing, Image Processing and Pattern Recognition
Vol.9, No.4 (2016), pp.369-392
http://dx.doi.org/10.14257/ijsip.2016.9.4.33

Auto-Generation Method of Butterfly Pattern of Batik Based on Fractal Geometry

Qingni Yuan, Jian Lv and Haisong Huang

Key Laboratory of Advanced Manufacturing Technology (Guizhou University), Ministry of Education, Guiyang 550003, China.
cme.qnyuan@gzu.edu.cn

Abstract

Batik has important aesthetic and cultural values, which records and expresses national culture. Their basic graphical pattern comes from imitating object in natural environment that lots of patterns have self-similar. In this paper, we first analyze self-similar characteristic of butterfly pattern Batik and propose an auto-generation algorithm of butterfly pattern based on fractal geometry. The auto-generation algorithm adopts two-dimensional iterated function system (IFS) to construct fractal. First, it generates basis butterfly shape element by functions and it then repeatable iterates the functions until forming a butterfly pattern. Second, it generates butterfly beard to obtain an entire butterfly pattern by functions. Finally, it uses function to iterate the entire butterfly pattern to generate various pattern layouts. The experimental result shows that the algorithm can automatically generate various and beautiful butterfly fractal pattern. Based on the experimental result, we analyze change regularity between generation patterns and function parameters to obtain the most effective range of various function parameter values, which can perfect the proposed algorithm to quickly generate effective patterns. It realizes digital design of butterfly pattern and enriches pattern of Batik.

***Keywords**: Auto-generation, Iterated Function System, Fractal Geometry, Butterfly Pattern, Batik*

1. Introduction

Batik is one of folk traditional dyeing crafts. It has both aesthetic and cultural heritage values, so it occupies an important position in the history of modern textile in the world. Graphical design of Batik typically carries a reusing by variety of basic graphical shape/pattern (see Figure 1). The basic graphical shape is obtained from the imitation of natural environment object, which records and expresses national culture. It mainly includes animal shape, plant shape and geometric shape. Butterfly is one of the best main imitating animal shapes, because butterfly shape represents the happiness life and the children reproduction.

Tradition production method of Batik is to paint pattern on the cloth by handmade and then dip dyeing in blue dyestuff. This way is to be low production rate and disadvantage of protection of intangible cultural heritage. With the development of computer technology, digital dyeing is a new production method, which is digital from graphical design to dyeing by computer. It not only realizes the digital protection of tradition Batik, but also greatly promotes the development of digital art and design technology of Batik.

ISSN: 2005-4254 IJSIP

Figure 1. Graphical Design of Batik

Digital design technology is automatic generation pattern by computer, which greatly enrich and develop graphical design of tradition Batik. According to Yan Yan Sunarya [1], Batik motifs as well as zeolite molecular structures share similarity in its richness of variants and modules. Those aspects become the potential resource to elaborate new Batik that applies molecular zeolite structures and state of origins. His research is conducted through form morphology and adaptation analysis to find its similarity as the benchmarking for contemporary batik zeolit motifs and design development system. Ayung Candra Padmasari [2] proposes interior Batik gallery using normal mapping for virtual reality. Yulianto [3] proposes to create Batik motive rules based on fractal, which a morphological erosion process is generated to create template, while repetition and pattern placement are conducted under conformal fractal transformation based on iterated function system (IFS). IFS also conduct extensive restriction to object formed [4]. The research in the field of fractal theory is a hotspot. It is good at the growth model such as Diffusion Limited Aggregation model (DAL) [5-8], Lindenmayer system (L-system) [9-12]. DLA is the process whereby particles undergoing a random walk due to Brownian motion cluster together to form aggregates of such particles. The clusters formed in DLA processes are referred to as Brownian trees. These clusters are an example of fractals. L-systems are used to model the morphology of a variety of organisms and can be used to generate self-similar fractals such as iterated function systems (IFS). In the paper, we proposed an automatic generation method of butterfly pattern of Batik based on iterated function systems (IFS) of fractal. We can realize the simulation of butterfly pattern by generating self-similar fractals.

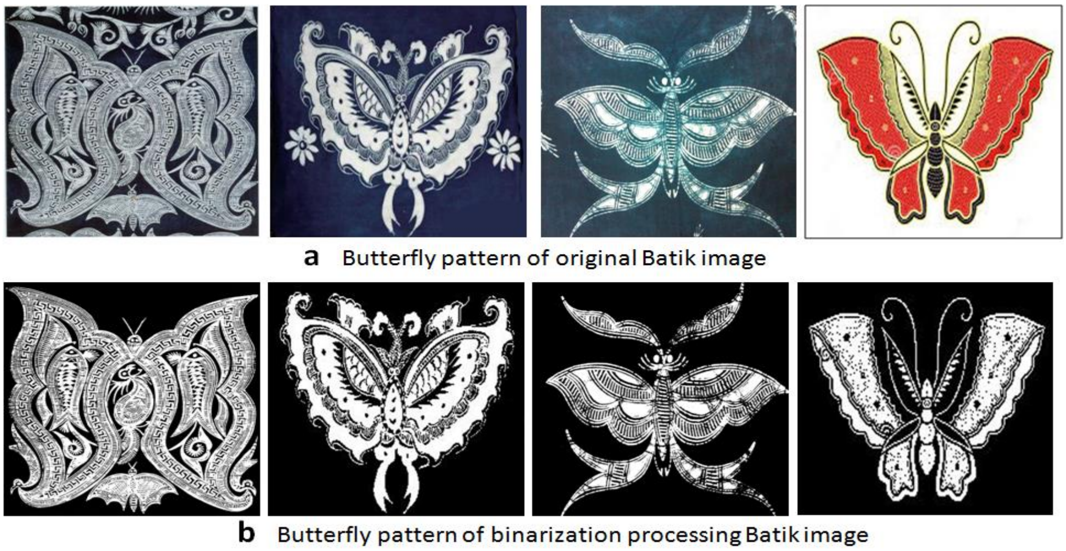


Figure 2. Butterfly Pattern of Batik

2. Visual Characteristics Analysis of Butterfly Pattern of Batik

Traditional Batik is a technology of wax-resist dyeing applied to whole cloth, or cloth made using this technology. Patterns are drawn with pencil and later redrawn using hot wax, usually made from a mixture of paraffin or bees wax, which functions as a dye-resist. The wax can be applied with a variety of tools.

Batik patterns are mainly symbolic. The pattern of butterfly is largely depicted on Batik with symbols designed to bring lots of child. From Figure 2, we can see the butterfly pattern of Batik has self-similar characteristics. That is, these butterfly pattern exhibit a repeating pattern that displays at every scale and the replication is exactly identical at every scale. It is a self-similar pattern. Because fractals can also be nearly identical at different levels and also include the idea of a detailed pattern that repeats itself, fractal geometry can be applied to butterfly pattern design of traditional Batik. Besides, from Figure 1, we can see that the layout of Batik pattern is mainly symmetrical, double-square, four-square, and circle, which have self-similar and self-affinity characteristics of fractal geometry.

3. Fractal Geometry

A fractal is a mathematical set that exhibits a repeating pattern that displays at every scale. If the replication is exactly identical at every scale, it is called a self-similar pattern. Fractals also include the idea of a detailed pattern that repeats itself [13]. The term 'fractal' was first used by mathematician B.B.Mandelbrot in 1975 [14-15]. In mathematics, iterated function system (IFS) is a method of constructing fractals; the resulting constructions are always self-similar. IFS fractals can be of any number of dimensions, but are commonly computed and drawn in 2D. The fractal is made up of the union of several copies of itself, each copy being transformed by a function (hence 'function systems').

A two-dimensional IFS is a finite collection of n function Fi from R2 to R2. The solution of system is the set S in R2 (and hence an image) that is the fixed point of Hutchinson's recursive set equation [16]:

$$S = \bigcup_{i=0}^{n-1} F_i(s) \tag{1}$$

As implemented and popularized by Barnsley [17], linear technically they are affine as each is a two by three matrix capable of expressing scale, rotation, translation:

$$F_i(x, y) = (a_i x + b_i y + c_i, d_i x + e_i y + f_i) \tag{2}$$

In order to facilitate the proofs and guarantee convergence of the algorithms, the functions are normally constrained to be contractive, that is, to bring points closer together.

4. Auto-Generation Algorithm of Butterfly Pattern

The auto-generation algorithm of butterfly pattern is based on two-dimensional IFS to construct fractal. First, it generates basis shape element of butterfly by functions and it then repeatable iterates the functions until forming a butterfly pattern. Second, it generates butterfly beard to obtain an entire butterfly pattern by functions. Finally, it uses function to iterate the entire butterfly pattern to generate pattern layout. According to the iteration number, it automatically generates the butterfly pattern layout that is symmetrical layout, double-square layout, or four-square layout, *etc*. Figure 3 shows the algorithm flow.

4.1. Generating Basis Shape Element of Butterfly

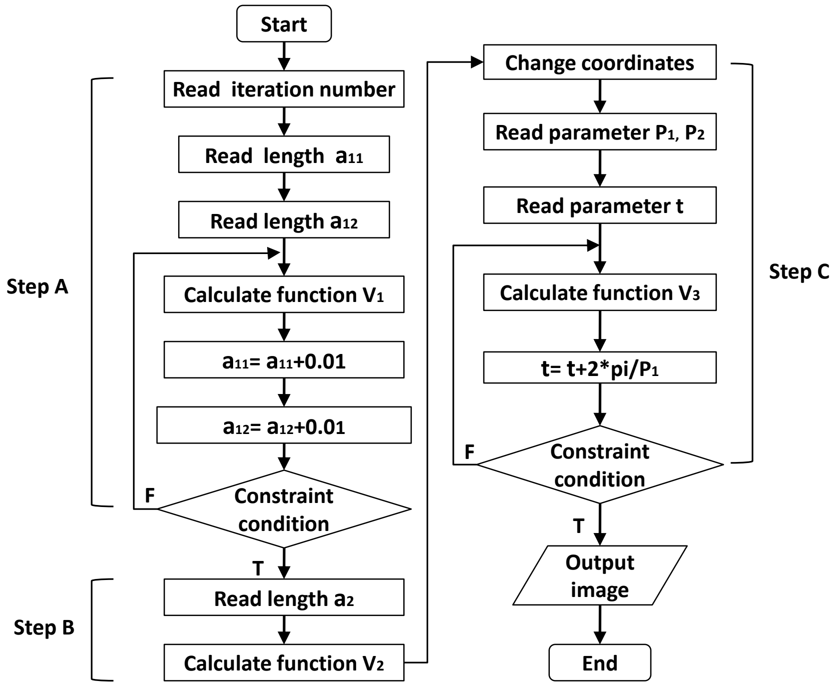


Figure 3. Auto-Generation Algorithm Flowchart

Basis shape element of butterfly to be generated is based on Rhodonea curve or rose that is a sinusoid plotted in polar coordinates. Up to similarity, these curve can all be expressed by a polar equation of the form [18-19]

$$\rho = \cos(n\theta) \tag{3}$$

or, alternatively, as a pair of Cartesian parametric equation of the form

$$\begin{cases} x = \cos(nt)\cos(t) \\ y = \cos(nt)\sin(t) \end{cases} \tag{4}$$

if n is an integer, the curve will be rose-shaped with 2n petals if n is even, and n petals if n is odd (see Figure 4). When n is even, the entire graph of the rose will be traced out exactly once when the value θ change from 0 to 2π . When n is odd, this will happen on the interval between 0 and π . Since

$$\sin(n\theta) = \cos(n\theta - \frac{\pi}{2}) = \cos(n(\theta - \frac{\pi}{2n})) \tag{5}$$

For all θ, the curves given by the polar equations $\rho = \sin(n\theta)$ and $\rho = \cos(n\theta)$ are identical except for a rotation of $\pi/2n$ radians.

A rose whose polar equation is of the form

$$\rho = a\cos(n\theta) \tag{6}$$

where n is a positive integer, has area

$$\frac{1}{2}\int_0^{2\pi}(a\cos(n\theta))^2 d\theta = \frac{a^2}{2}(\pi + \frac{\sin(4n\pi)}{4n}) = \frac{\pi a^2}{2} \tag{7}$$

if n is even , and

$$\frac{1}{2}\int_0^{\pi}(a\cos(n\theta))^2 d\theta = \frac{a^2}{2}\left(\frac{\pi}{2}+\frac{\sin(2n\pi)}{4n}\right)=\frac{\pi a^2}{4} \tag{8}$$

if n is odd. The same applies to roses with polar equations of the form

$$\rho = a\sin(n\theta) \tag{9}$$

where a is length of petal.

Thus, we define a non-linear function V_1 to generate the basis shape elements of butterfly

$$V_1 = \begin{cases} a_{11}\cos(2\theta+\frac{\pi}{2}), \theta\in[0,\pi] \\ a_{12}\cos(2\theta-\frac{\pi}{2}), \theta\in[\pi,2\pi] \end{cases} \tag{10}$$

Where the parameter value θ change from 0 to π depict up part of the basis shape element of butterfly, and the parameter a11 is its length size. The parameter value θ change from π to 2π depict down part of the basis shape elements of butterfly, and the parameter ais its length (see Figure 4).

Then, we perform repeated iteration based on the function V_1. We call each such function V_{1j} a variation. The initial variations were simple remapping of the plane. They were followed by dependent variations, where the coefficients of the affine transform define the behavior of the variation. Variations are controlled by additional parameters independent of the affine transform.

Variations can be further generalized by replacing the integer parameter j with a blending vector vij with one coefficient per variation. Then

$$F_{1i}(\rho,\theta)=\sum_j v_{ij}V_{1j}(\rho,\theta) \tag{11}$$

Where the iterated parameter i is a random integer from 0 to n-1. With this generalization, we obtain a butterfly pattern $F_1(\rho,\theta)$.

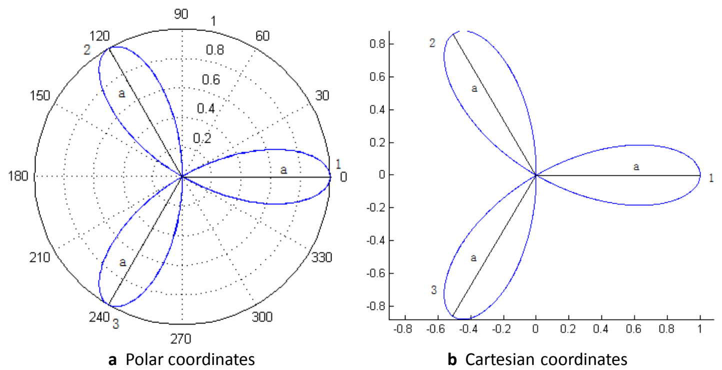


Figure 4. Three Petals Rose Curve

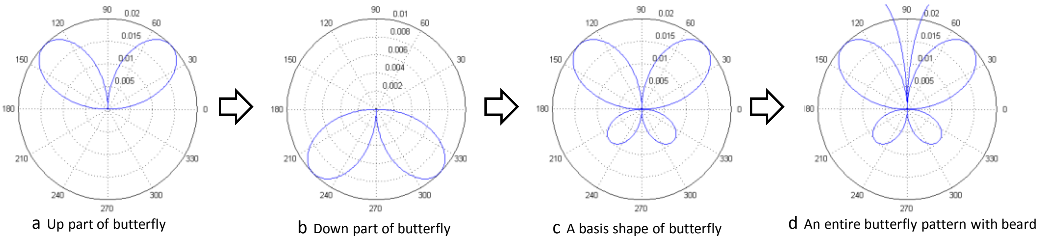
4.2. Generating Beard of Butterfly

From above step, we have a butterfly pattern. Now, we define a non-linear function V_2 to generate beard base on the butterfly pattern

$$V_2 = \begin{cases} a_2\sin(6\theta), \theta\in[\frac{18\pi}{12},\frac{19\pi}{12}] \\ a_2\sin(6\theta+\pi), \theta\in[\frac{17\pi}{12},\frac{18\pi}{12}] \end{cases} \tag{12}$$

Where the parameter value θ change from $18\pi/12$ to $19\pi/12$ depict left beard of the butterfly; and the parameter value θ change from $17\pi/12$ to $18\pi/12$ depict right bread of the butterfly (see Figure 5). Thus, we integrate the breads and a butterfly pattern $F_1(\rho,\theta)$ to obtain an entire butterfly pattern $F_2(\rho,\theta)$,

$$F_2(\rho,\theta) = F_1(\rho,\theta) + V(\rho,\theta) \tag{13}$$



a Up part of butterfly b Down part of butterfly C A basis shape of butterfly d An entire butterfly pattern with beard

Figure 5. Generation Process of a Basis Butterfly Shape

4.3. Generating Pattern Layout of Butterfly

For an entire butterfly pattern, we firstly change polar coordinates $F_2(\rho,\theta)$ into Cartesian coordinates $F_2(x,y)$. Second, we define a non-linear function V_3 that is called JuliaN variation. To set parameter P1=juliaN.power and P2=juliaN.dis

Where the power law is distributed density. The dist is size of distance. Now, the a non-linear function V3 is by

$$V_3(x,y) = r^{p_2/p_1} \cdot (\cos t, \sin t) \tag{14}$$

Where $t = \phi + \dfrac{2\pi}{p1}$, $\phi = \arctan(\dfrac{y}{x})$, $r = \sqrt{x^2 + y^2}$

Finally, we perform iteration by the function V3 based on an entire butterfly pattern of Cartesian coordinates $F_2(x,y)$. We define the function $F_3(x,y)$ as follows:

$$F_{3i}(x,y) = \sum_j v_{ij} V_{3j}(x,y) \tag{15}$$

Where the parameter V_{ij} is a blending vector with one coefficient per variation. The iterated parameter i is a random integer from 0 to n-1.

The function $F_3(x,y)$ is always applied regardless of the parameter value i in the iteration loop. The final transform is like a non-linear camera. The result of the application of final transform function is not 'in the loop', and there can be only final transform per an entire butterfly pattern $F_2(x,y)$. The final transform can have a post transform $F_2(x,y)$ associated with it.

The function $F_3(x,y)$ generates pattern layout of butterfly. According to the iteration number i, we obtain different layout pattern $F_{3_i}(x,y)$ such as: symmetrical layout, double-square layout, or four-square layout, *etc*.

5. Experiment and Analysis

The auto-generation algorithm of butterfly pattern of Batik is implemented in MATLAB R2014. This experiment generates various butterfly patterns and pattern layouts by setting various different parameters of the algorithm, so we obtain the most effective range of various parameters value by analyzing the experimental results.

5.1. Setting Parameters To Generate Butterfly Patterns

In this experiment, we perform parameter setting to generate various basis shape elements and patterns of butterfly. We through a non-linear function V_1 to generate the basis shape elements of butterfly. From Equation (10), the parameter a_{11} and a_{12} impact the basis shape of butterfly. The parameter a11 controls up half part size of the basis butterfly shape; while the parameter a_{12} controls down half part size of the basis butterfly shape. Figure 6 shows various the basis shape by setting parameters: $a_{12}=0.2a_{11}$, $a_{12}=0.3a_{11}$, $a_{12}=0.35a_{11}$, $a_{12}=0.35a_{11}$, $a_{12}=0.4a_{11}$, $a_{12}=0.5a_{11}$, $a_{12}=0.618a_{11}$, $a_{12}=0.7a_{11}$ and $a_{12}=0.8a_{11}$.

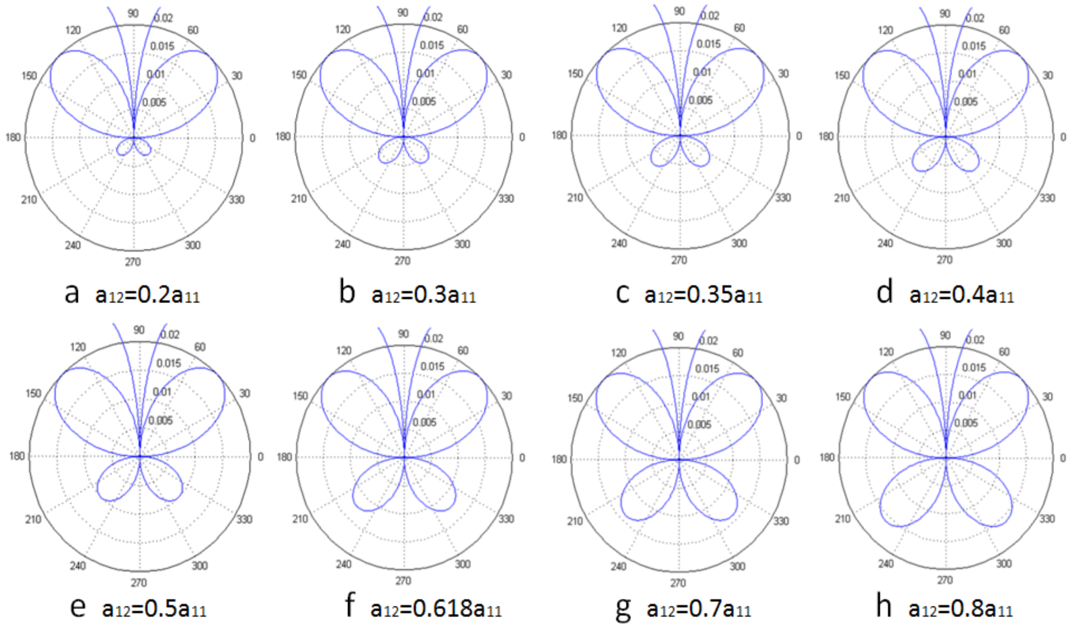


Figure 6. Generation Various Basis Butterfly Shape By Setting Parameter a_{11} and a_{12}

From Figure 6 , we see that the figure 6 from b to g are best similar with butterfly, so we adopt that scale value of parameter a11 and a12 is range from 0.3 to 0.7 in our auto-generation algorithm. Next, we adopt the basis butterfly shape elements with $a_{12}=0.3a_{11}$ and $a_{12}=0.618a_{11}$ to perform experiment to generate various butterfly patterns.

From Equation (11), a butterfly pattern is generated by repeated iteration based on a basis butterfly shape element. Parameter i (iteration number) impacts the generation shape, so we set different parameter value of i to generate various butterfly pattern. In the implementation, parameter value of i is from 1 to 20; while line style is solid line. In Figure 7 with the basis shape $a_{12}=0.3a_{11}$, it shows various butterfly patterns with different parameter value i=2,3,5,6,11,15 and different line style: (1) From a1 to a6 are square, (2) From b1 to b6 are integration of solid line and diamond, (3) From c1 to c6 are integration of solid line and circle, (4) From d1 to d6 are integration of solid line and hexagram, (4) From e1 to e6 are integration of solid line and asterisk. In Figure 8 with the basis shape $a_{12}=0.618a_{11}$, it shows various butterfly patterns with different parameter value i=2,3,5,6,11,15 and different line style: (1) From a1 to a6 are solid line, (2) From b1 to b6 are circle, (3) From c1 to c6 are asterisk, (4) From d1 to d6 are diamond, (4) From e1 to e6 are plus sign.

From Figure 7 and 8, we see that there are various experimental result when a basis shape element is iterated by different parameter values i and different line styles. In our implementations, when iteration parameter i is more than 15, the generation butterfly pattern is very similar. Thus, we adopt that range value of iteration parameter i is from 2 to 15.

5.2. Setting Parameters To Generate Pattern Layouts

In the section, we generate various butterfly pattern layouts based on above experimental results. Adopt. From Equation (14), various pattern layouts is automatic generated by setting parameter P_1 and P_2, which P_1 controls butterfly number/density in layout, and P_2 controls distance between these butterfly numbers. In the implementations, we adopt that value of parameter P_1 is from 2 to 6, and value of parameter P_2 is from 0.1 to 1.

Figure 9 shows pattern layouts with $P_1 = 2, 3, 4, 5$ and $P_2 = 1, 0.5$. In these pattern layouts, we first rotate angle 45 for a butterfly pattern of above experimental result, then we calculate the function $F_3 = (x, y)$ in Equation (15) to perform fractal, which the butterfly pattern is basis shape parameter $a_{12} = 0.618a_{11}$, iteration parameter i=15, and solid line style. Figure 10 shows pattern layouts.

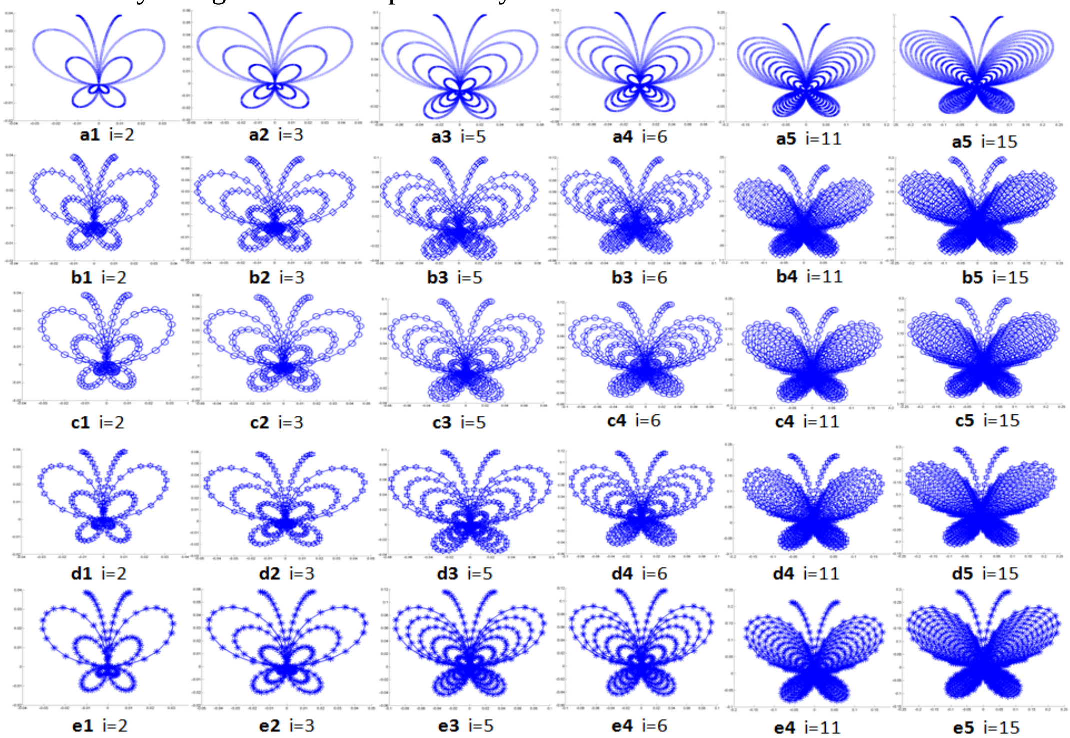


Figure 7. Generate Entire Butterfly Patterns By Setting Parameter i=2,3,5,6,11,15 with Basis Shape $a_{12} = 0.3a_{11}$

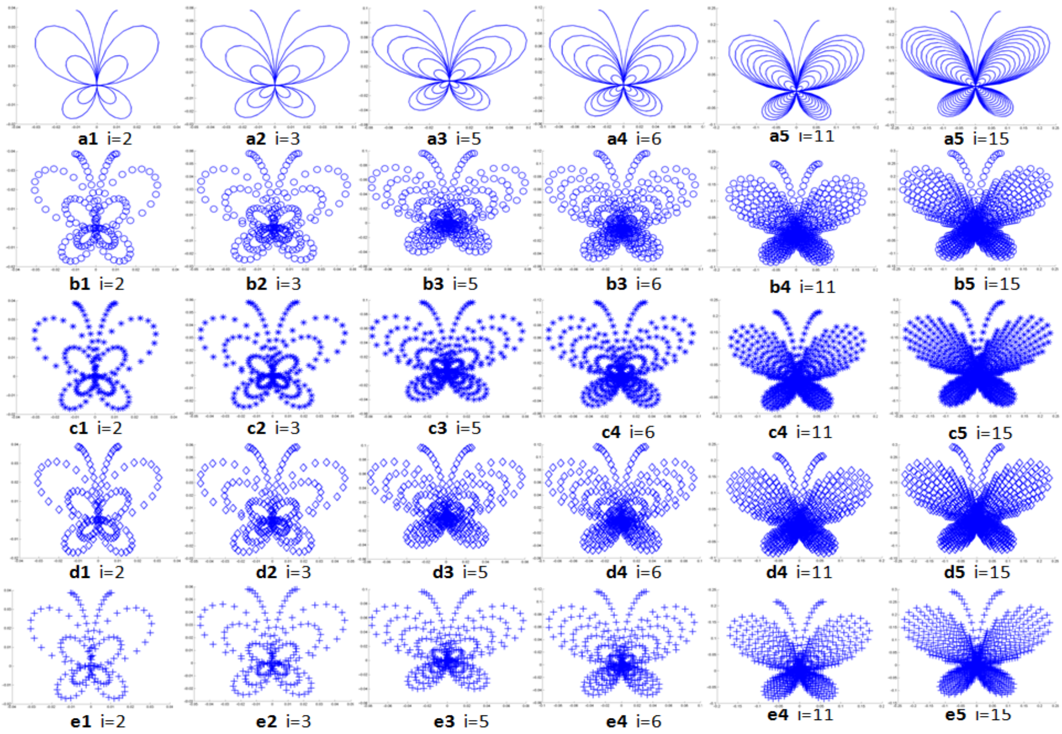


Figure 8. Generate Entire Butterfly Patterns By Setting Parameter i=2,3,5,6,11,15 with Basis Shape $a_{12} = 0.618a_{11}$

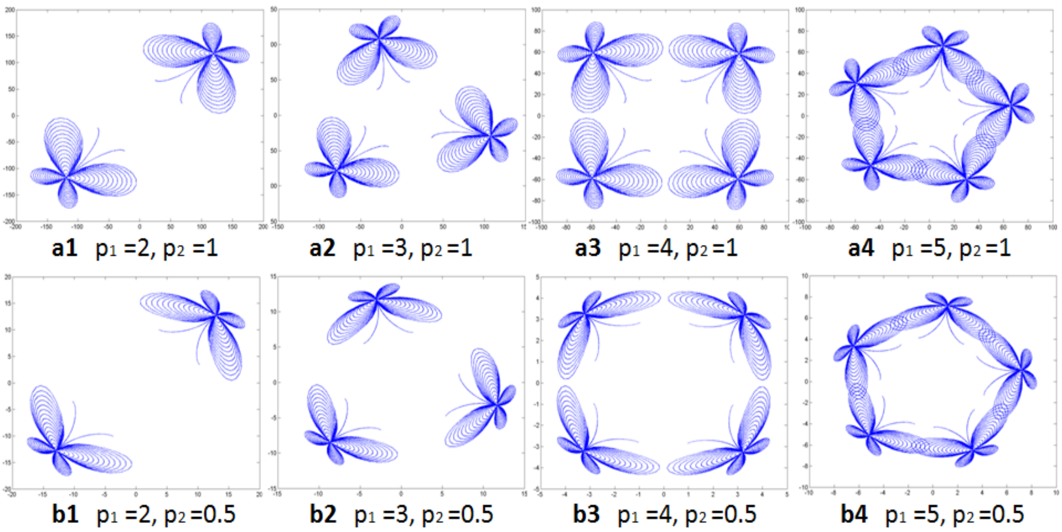
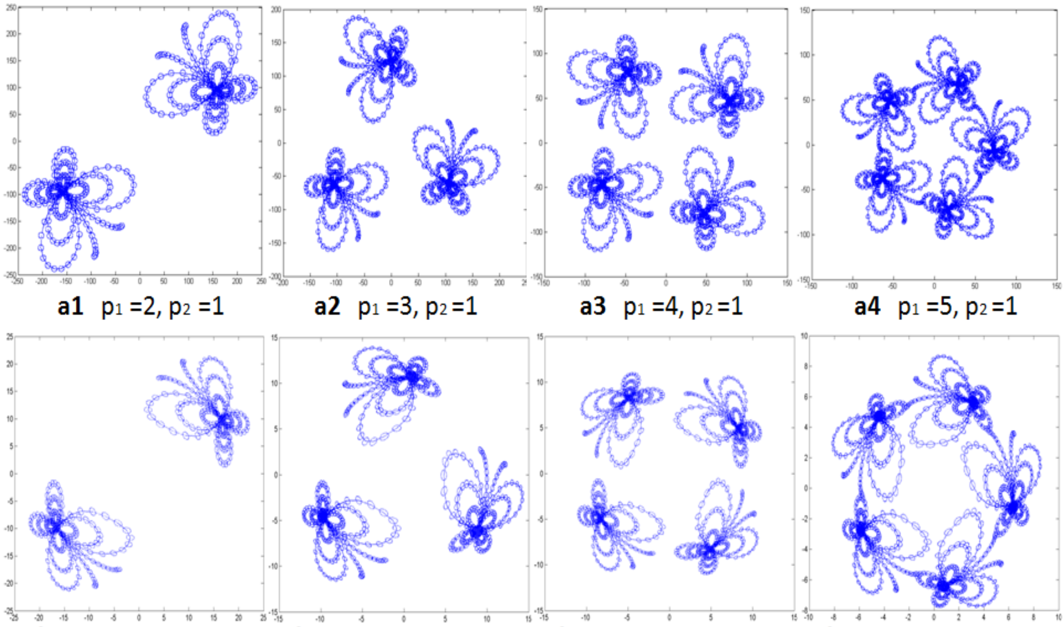


Figure 9. Generate Pattern Layouts By Setting Parameter P_1=2,3,4,5 and P_2=1, 0.5

with $P_1 = 2, 3, 4, 5$ and $P_2 = 1, 0.618$. In these pattern layouts, we do not rotate the butterfly pattern of above experimental result, which the butterfly pattern is basis shape parameter $a_{12} = 0.3a_{11}$, iteration parameter i=3, and integration line style of solid line and circle.

From Figure 9 and 10, we see that parameter value P_1 represents butterfly numbers and layout characteristic: the parameter $P_1 = 2$ is two butterflies and symmetrical layout, the parameter $P_1 = 3$ is three butterflies and three-circle layout, the parameter $P_1 = 4$ is four butterflies and four-square layout, the parameter $P_1 = 5$ is five butterflies and five-circle layout. Besides, the parameter P_2 represents change distance of each butterfly pattern.



a1 $p_1 = 2, p_2 = 1$ a2 $p_1 = 3, p_2 = 1$ a3 $p_1 = 4, p_2 = 1$ a4 $p_1 = 5, p_2 = 1$

b1 $p_1 = 2, p_2 = 0.618$ b2 $p_1 = 3, p_2 = 0.618$ b3 $p_1 = 4, p_2 = 0.618$ b4 $p_1 = 5, p_2 = 0.618$

Figure 10. Generate Pattern Layouts By Setting Parameter P_1=2,3,4,5 and P_2=1, 0.618

Now, we generate basic framework of butterfly Batik after two iterating calculation based on functions, which first iterating calculation generates a butterfly pattern based on the function V_1; second iterating calculation generates butterfly pattern layout based on the function V_3. Traditional butterfly Batik has various ornaments with it, such as flower, plant, leaf, *etc*. We also may decorate some flower or plant into our auto-generation butterfly pattern, which can obtain finally and beautiful Batik pattern. We then output it to render that applies color to it, which can have the same beautiful result with tradition Batik.

Figure 11. shows the finally and beautiful butterfly pattern with various ornament flowers. The final butterfly pattern is four-square butterfly pattern layout that is often applied in tradition Batik. These flowers are automatic generated by rose curve. Each flower integrates various size functions of rose curve, and repeatedly iterates these functions to generate them, so each flower is also fractal pattern.

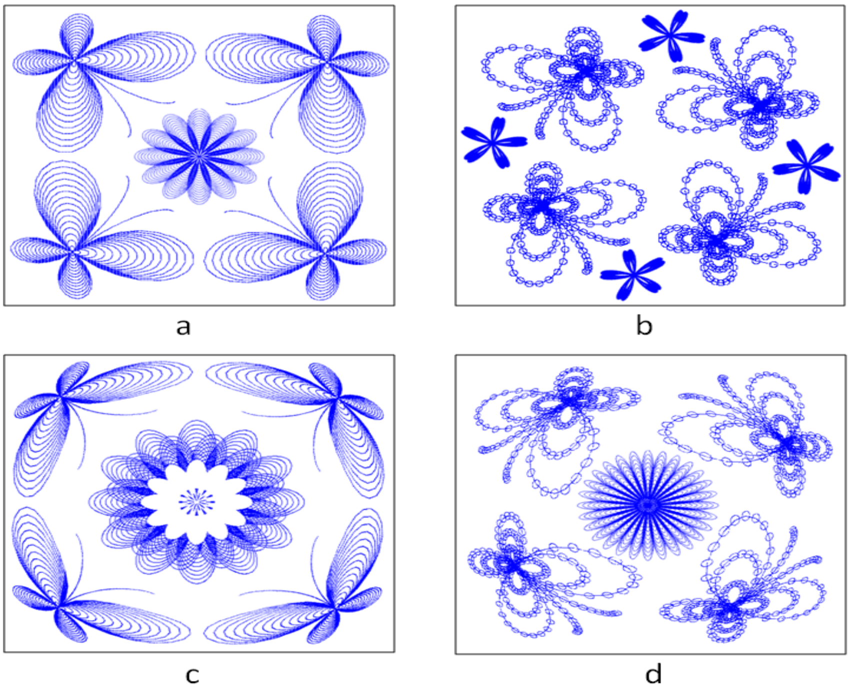


Figure 11. Generate Final Butterfly Pattern with Flower

6. Conclusions and Future Work

In the paper, we proposed a new auto-generation algorithm of butterfly pattern of Batik based on IFS of fractal geometry. It first generates a basis butterfly shape based on defined function that is rose curve. Second, it performs repeated iteration based on the function to generate a butterfly pattern. Finally, it again performs iterated function based on the butterfly pattern to generate various pattern layouts. Because different parameter values have different patterns or shapes that is not very like, we researched change regularity between shape and parameter to quickly generate effective shape based on our experimental result. By our research, it realizes digital design of butterfly pattern Batik. It is not only basis of digital production, but also can convenient for digital protection of Batik. It is noted that the proposed method is only applied self-similar pattern in Batik, and need to use another renderer to achieve as effect as traditional Batik. In the future, we extend this algorithm to apply to more pattern design, which enrich digital design patterns of Batik.

Acknowledgements

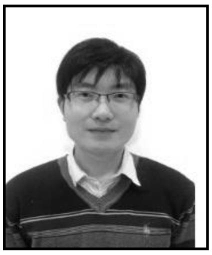
This research is supported by National Science and Technology Support Program Mission Statement for " Tourism industry demonstration for digitization of cultural heritage, cultural protection and development of minorities Features "(2014BAH05F00), Guizhou Province Science and Technology Office Project for " Key Laboratory of Construction of Digital Cultural Heritage Protection in Guizhou Province " (Guizhou Branch Total Z [2013] 4005), Guizhou Province Science and Technology Fund Project (Guizhou Branch Total LH [2014] No. 7645,7644). Guizhou Province Science and Technology Fund Project (Guizhou Branch Total JZ [2014] No. 2004; Guizhou Branch Total [2015] No. 13).

References

[1] Y. Y. Sunarya, A. Z. Mansoor and D. Widiawati, “Re-inventing Batik: zeolite engineered pore structure as resource of contemporary Batik system”, [J]. International Journal of Information Technology and Business Management, vol. 9, no. 29, **(2014),** pp. 44-48.
[2] A. C. Padmasari, C. Christyowidiasmoro and M. Hariadi, “Interior Batik Gallery Using Normal Mapping For Virtual Reality”, [J]. IPTEK Journal of Proceedings Series, **(2014)**, vol. 1, pp. 374-379.
[3] Y. Rusmono and H. Mochamad, “Iterative function system algorithm based on conformal fractal transformation for Batik motive design”, [J]. Journal of Theoretical & Applied Information Technology, vol. 10, no. 62, **(2014)**, pp. 275-280.
[4] A. L. Barabási and H.E.Stanley, “Fractal Concepts in Surface Growth”, [M].Cambridge University Press, **(1995)**.
[5] W. Guo and R. Zhang, “A new method to simulate fancy yarn based on DLA model”[J]. Application of Computer System, vol. 5, no. 19, **(2010)**, pp. 46-74.
[6] A. Asselah and A. Gaudilliere, “From logarithmic to subdiffusive polynomial fluctuations for internal dla and related growth models”, Arxiv, preprint arXiv:1009.2838, **(2010)**.
[7] C. Lucas, “The limiting shape for drifted internal diffusion limited aggregation is a true heat ball”,[J]. Probability Theory and Related Fields, vol. 6, no. 159, **(2014)**, pp.197-235.
[8] D. Li, W. Yanting and Q. Zhongcan, “Diffusion limited aggregation with polygon particles”, [J]. Communication in Theoretical Physics,vol. 11, no.15, **(2012)**, pp. 895-901.
[9] https://en.wikipedia.org/wiki/L-system.
[10] J. B. Bassingthwaighte, L. S. Liebovitch and B. J. West, “Fractal Physiology”,[M]. Oxford University Press, **(2013)**.
[11] Affiliations,”Fractal atomic-level percolation in metallic glasses”,[J]. Science Magazine, vol. 9, no. 18, **(2015)**, pp. 1306-1310**.**
[12] Y. Q. Cheng and E. Ma, “ Atomic-level structure and structure-property relationship in metallic glasses. Prog. Mater. Sci., no. 56, **(2011)**, pp. 379-473.
[13] J.-F. Gouyet, “Physics and fractal structures”, [M]. Paris/New York: Masson Springer, 1996. ISBN 978-0-387-94153-0.
[14] B. B. Mandelbrot, “The fractal geometry of nature”, [M]. Macmillan, 1983. ISBN 978-0-7167-1186-5. Retrieved, 1 February **(2012)**.
[15] K. Falconer, “Fractal Geometry: Mathematical Foundations and Applications”, John Wiley & Sons, Ltd., 2003. ISBN 0-470-84862-6, **(2003).**
[16] J. Hutchinson, “Fractals and Self-Similarity”, Indiana University Journal of Mathematics, no. 30, **(1981)**, pp. 713-747.
[17] M. Barnsley, “Fractals Everywhere”, Academic Press, San Diego, **(1988)**.
[18] J. J. O’Connor and E. E. Robertson, "Rhodonea", MacTutor History of Mathematics archive, University of St Andrews.
[19] H. M. Cundy and A.P. Rollett, “Mathematical Models second edition”, (Oxford University Press), **(1961)**, pp. 73.

Authors

Yuan Qing Ni, female, born in 1976, Guizhou Weng'an, Associate Professor of Key Laboratory of Advanced Manufacturing Technology, Guizhou University, PhD, Master Instructor, major research direction is digital design and manufacturing. Published nearly 10 papers, 8 of them are included in SCI / EI / IS.



Lv Jian, male, born in November, 1983, Professor of Key Laboratory of Advanced Manufacturing Technology, Guizhou University, PhD, Master Instructor, major research directions are creation design, industry design, *etc*. Published 5 papers, 3 of them are included in SCI / EI / ISTP.

Huang Haisong, female, born in October, 1977, Guizhou Dafang, Professor of Key Laboratory of Advanced Manufacturing Technology, Guizhou University, PhD, Master Instructor, deputy director, major research directions are advanced manufacturing technology and artificial intelligence. Published nearly 10 papers, 5 of them are included in SCI / EI / ISTP.