

Regular paper

Research on National Pattern Reuse Design and Optimization Method Based on Improved Shape Grammar

Ning Ding, Jian Lv^{*}, Lai Hu*Key Laboratory of Advanced Manufacturing Technology, Ministry of Education, Guizhou University, Huaxi District, Guiyang City, Guizhou Province, China, 550025*

ARTICLE INFO

Article History

Received 18 Jul 2019

Accepted 03 Mar 2020

Keywords

Shape grammar

Batik patterns

Optimization design

Pattern reuse

Pattern reuse design system

ABSTRACT

Considering the low degree of abstraction of traditional shape grammar in national pattern reuse design, this paper proposes a method based on the combination of improved shape grammar and an optimization algorithm to reuse national patterns design. As an application example, we carried out research on the batik pattern elements of the Miao nationality in the “Intangible Cultural Heritage of Guizhou Province” system. Firstly, an improved shape grammar based on synthetic predicate grammar coding is developed to represent the transformation process of design pattern reuse in batik patterns. Then, the number of patterns arranged in a specific position is taken as the optimization variable; the particle swarm optimization algorithm is used to optimize the variables, so as to obtain the optimized predicate shape grammar parameters; and the pattern is reused according to the parameters. A batik design pattern reuse system is also constructed to realize rapid design. The results show that the proposed method can enrich the reuse results of batik pattern elements and realize the rapid design of complex batik patterns.

© 2020 The Authors. Published by Atlantis Press SARL.

This is an open access article distributed under the CC BY-NC 4.0 license (<http://creativecommons.org/licenses/by-nc/4.0/>).

1. INTRODUCTION

National patterns design has an element of fuzziness and uncertainty to them that is related to subjective psychology and image perception [1]. Traditional pattern design is subject to the functional and aesthetic demands of manual design, resulting in low design efficiency and strong subjectivity. However, in recent decades, with the development of computer graphics, machine learning, and the combination of digital and graphic design, computer-aided design (CAD) technology has been able to help designers meet their design requirements [2,3]. Shape grammar, with its grammar rule base, recursive operation mechanism, and intuitive visualization, can explain design well through formal description. In addition, it can creatively support designers to better understand and perceive design and to promote the development of CAD tools, making design more scientific, reliable, and universal [4–6].

In the literature, research on combining shape grammar with other optimization algorithms focuses on combining shape grammar with parallel particle swarm optimization (PSO), which was proposed by Keles [7], in order to solve the problem of embedding and replacing the emergent part after the design is generated with shape grammar. Chau *et al.* [8] proposed a method that combines shape grammar with lattice theory and uses lattice-based rational quadratic Bézier curves to represent three-dimensional circular arc shape grammar, which solves the complexity of geometric operations related to specific syntactic shapes in shape computation. Jia and Ming [9] proposed the dynamic shape representation based on

shape grammar and extend the theory of shape grammar and further enriches the form of grammar rules. To improve traditional shape grammar, Abdelbar and Rachid [10] proposed a parametric shape grammar method to generate a template motif. Jia and Ming [11] proposed hierarchical shape grammar and used descriptive language concepts to apply shape grammar to the creative design of Zhuang embroidery patterns. Zahra *et al.* [12] proposed the parametric shape grammar design. Although traditional shape grammar has great advantages in design evolution, for the art of graphic design, the degree of abstraction of traditional shape grammar is low. Therefore, for the innovative design of national patterns, the traditional and single use of shape grammar cannot adequately meet the needs of pattern reuse and rapid design.

Therefore, based on the above analysis and considering the problem of low abstraction and reuse of national pattern design, the main contributions of this study are as follows:

- An improved shape grammar of comprehensive predicate grammar is developed to represent the generation process of national pattern redesign.
- Taking the Miao batik pattern as the application carrier, the number of pattern elements in the pattern is taken as the optimization basis for the first time, the mathematical model is established, and the pattern arrangement parameters are optimized by using the PSO algorithm, so as to obtain the optimized predicate shape grammar parameters and realize the reuse and design generation of pattern elements in the Miao batik pattern. Improved shape grammar is introduced into the design system to realize the rapid design of different patterns.

^{*}Corresponding author. Email: bwppwen@126.com

- Pattern elements are obtained that captured high user attention, as demonstrated by an eye movement experiment.

2. RESEARCH BACKGROUND

2.1. Basic Definitions of Shape Grammar

Shape grammar is a common method for pattern design, originally proposed by Stiny and Gips in 1972 [13]. This grammar provides a formal mechanism to support design space exploration by using visual replacement rules and defines shape language by an algorithm, which makes the transformation of shape have the characteristics of label and parameter. The application of shape grammar rules is divided into two steps: (1) sub-shape detection. That is, identifying a specific shape, and (2) replacing it with another shape.

Stiny defined shape grammar as a quadruple, that is, $G = \langle S, R, L, I \rangle$, where S is a finite set of shapes s , R is a regular finite set of r , L is a finite set of marks l , $\langle S, L \rangle^+$ is a nonempty set of all marker properties consisting of all shapes in the shape set S and symbols in the marker set L , $\langle S, L \rangle^*$ is a set containing all marker shapes in the shape set $\langle S, L \rangle^+$ and empty marker shape set $\langle s_\emptyset, \emptyset \rangle$, I is an initial finite shape set, and $I \in \langle S, L \rangle^+$.

When there is an Euclidean transformation τ , the shape transformation rules r are generated in the form of $\alpha \rightarrow \beta$, where the shape to the left of the shape rule is represented by α , and $\alpha \in \langle S, L \rangle^+$. The shape to the right of the shape rule is represented by β , and $\beta \in \langle S, L \rangle^*$. So, shape rules r can be understood as functions with input shapes α and output shapes β : $r(\alpha) = \beta, r \in R$. To the initial shape, the expression that applies the shape rule to the shape is $[s - \tau(\alpha)] + \tau(\beta)$, where $\tau(\alpha)$ is the Euclidean transformed α , and $\tau(\alpha) \leq s$, and $\tau(\beta)$ is β subjected to the same transformation.

By applying the shape rules, new shapes can be continuously generated from the initial shapes, and shape rules are sequentially applied to the obtained shapes; that is, the shape sequence s_0, s_1, \dots, s_n where s_0 is the initial shape, s_{i+1} is the shape obtained from the s_i application of the shape rules ($i = 0, 1, \dots, n$), and each generated shape forms a shape language defined by the shape rules [1].

Based on the abovementioned properties, shape grammar can be regarded as a kind of visualized mathematics. This grammar can abstract the concrete shape into abundant and constructive descriptions, which is convenient for computational implementation and forms the basis of visual computation.

2.2. Euclidean Transformation

Since this study only focuses on the variation of the state model in the two-dimensional space, we only consider Euclidean transformation in the two-dimensional space. That is, Euclidean plane is regarded as a set of points satisfying special relations expressed by distance and angle. A new shape is created by varying the displacement, angle, scale or a limited combination thereof of the initial shapes [14]. The Euclidean distance formula is given in Equation (1):

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (1)$$

Figure 1 illustrates the definition of shape grammar and the application of Euclidean transforms in shape grammar.

The graph generated in Figure 1(c) was derived from the rules of Figure 1(a) and (b). That is, the final graph of Figure 1(c) is obtained by repeatedly applying the rule shown in Figure 1(a), starting from the initial shape shown in Figure 1(b). Each time, the rule uses a different Euclidean transformation to apply a combination of translation and rotation to the left-hand side of the shape rule, with the nested small diamond corresponding to the marker point l .

2.3. Particle Swarm Optimization

PSO was proposed by Russell Eberhart and James Kennedy in 1995. This algorithm uses a method inspired by the movement of birds or the gathering of fish to optimize a problem. This is accomplished by the direction of individual movement in the population, considering the optimal fitness of the population and its own optimal fitness to achieve the purpose of optimization, and by following the current search for the optimal value to find the global optimum [15,16].

In PSO, the potential solution of each optimization problem is a particle in the search space, each particle has a fitness value determined by an optimized function, and each particle has two main attributes: velocity and position. The position of the particle is a function-specific solution, and the velocity is used to determine the direction of the particle in the next iteration. Therefore, the particle updates itself by tracking two extremes in each iteration: (1) the optimal solution found by the particle itself “individual extremum” and (2) the optimal solution found so far in the entire population “global extremum.”

In a D-dimensional target search space, a colony is composed of N particles, wherein the formula for updating the velocity and the position of the i th particle is shown in Equations (2) and (3), respectively:

$$\vec{v}_i^{t+1} = \vec{v}_i^t + c_1 r_1 \times (\vec{bx}_i^t - \vec{x}_i^t) + c_2 r_2 \times (\vec{gbx}^t - \vec{x}_i^t) \quad (2)$$

$$\vec{x}_i^{t+1} = \vec{x}_i^t + \vec{v}_i^{t+1} \quad (3)$$

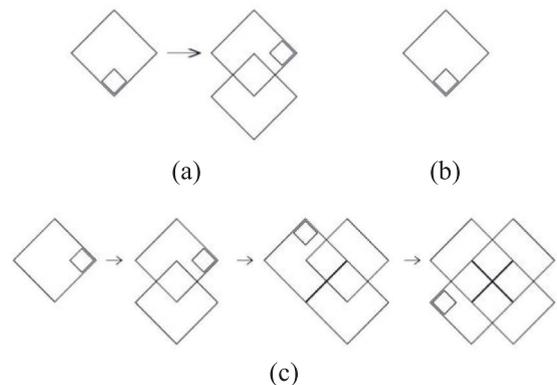


Figure 1 | Examples of traditional shape grammar applications. (a) Shape rules; (b) Initial shape; (c) Pattern derivation using shape grammar rules.

where c_1 and c_2 are the local and global acceleration constants, respectively; r_1 and r_2 are uniform random numbers in the range $[0, 1]$; t represents discrete time steps; \vec{v}_i^t is the velocity of particle i at time step t ; \vec{x}_i^t is the position vector of the i th particle (i.e., current solution) at time step t ; \vec{bx}_i^d is the best position vector of the i th particle up to time step t ; and \vec{gbx}^d is the best position of the all particles obtained up to time step t .

3. IMPROVED SHAPE GRAMMAR APPLICATION RULES

Traditional shape grammar uses marker points to control the application of shape rules in the design process. Although the sequential programming style can be well used to describe the design process by using marker points, it relies on specific shapes, which makes the design process too rigid [1]. Therefore, this study separate the shape knowledge and transformation process knowledge of patterns, focus on the application of transformation process knowledge; that is, the transformation process τ is regarded as a transformation verb (i.e., “predicate”), which concretizes the Euclidean transformation parameters and constitutes an improved shape grammar with comprehensive predicate grammar coding. The code of improved shape grammar rules is shown in Table 1.

In Table 1, τ represents all two-dimensional Euclidean transformation modes, 1/0 represents whether the initial shape is retained (1 is retained, 0 is not retained), $+\theta, -\theta$ represents clockwise or counterclockwise rotation about the center of the shape itself, (x_0, y_0) is the center of the circumferential array, N represents the number of arrays, $d_x, N_x; d_y, N_y$ represent the spacing and number in the X and Y directions, respectively, and *up, down, left, right* represent which boundary of the shape is mirrored as the axis of symmetry.

Use $V = \{(IS, \tau)\}$ to represent the application rule of improved shape grammar (IS represents the initial shape set) and call it the predicate grammar expression. The basic expression is shown in Equation (4), and the illustration is shown in Figure 2:

$$V = \left\{ \left(\begin{array}{c} is_1, Tran(d_x, d_y, 1/0) Rota(\pm\theta, 1/0) \\ Scal(s_x, s_y, 1/0) \end{array} \right) \right\}. \quad (4)$$

Table 1 Predicate shape grammar encoding table.

| Predicate | Meaning | Parameters | Range |
|-------------|----------------|--------------------------------------|---|
| <i>Tran</i> | Transformation | $d_x, d_y, 1/0$ | $Tran(v) \in \tau, \forall v \in R$ |
| <i>Rota</i> | Rotation | $\pm\theta, 1/0$ | $Rota(\theta) \in \tau, \forall v \in [0^\circ, 360^\circ]$ |
| <i>Scal</i> | Scaling | $s_x, s_y, 1/0$ | $Scal(u) \in \tau, \forall u \in R$ |
| <i>Circ</i> | Circular | $(x_0, y_0), N$ | - |
| <i>Lnar</i> | Line array | $d_x, N_x; d_y, N_y$ | - |
| <i>Mirr</i> | Mirror image | <i>up\down\left\right\right, 1/0</i> | - |
| \vdots | \vdots | \vdots | \vdots |
| { | start | - | - |
| } | stop | - | - |

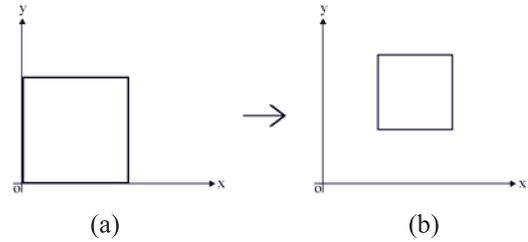


Figure 2 | A graphical representation of the basic predicate shape grammar. (a) Initial shape is_1 ; (b) Transformed graph s_τ .

The basic two-dimensional Euclidean transform matrix is expressed as

$$Tran(d_x, d_y) = \begin{bmatrix} x & y & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & d_x \\ 0 & 1 & d_y \\ 0 & 0 & 1 \end{bmatrix}$$

$$Rota(\theta) = \begin{bmatrix} x & y & 1 \end{bmatrix} \cdot \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$Scal(s_x, s_y) = \begin{bmatrix} x & y & 1 \end{bmatrix} \cdot \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

where d_x, d_y represent the coordinate variation of the shape s , and s_x, s_y represent the scaling ratio of the shape s , respectively.

The transformation process of Figure 3 is shown in Equation (5):

$$V = \left\{ \left(is_1, Tran(1, 1, 0) Scal\left(1/\sqrt{2}, 1/\sqrt{2}, 1\right) \right) \right\}. \quad (5)$$

If there is a circumferential array of shape transformation, that representation is defined as $V = \{(IS, (Circ(x_0, y_0), N))\}$. If the transformation form is a linear array, the definition representation is $V = \{(IS, Lnar(d_x, N_x; d_y, N_y))\}$. If the shape transformation is in the form of a mirror, the definition representation is $V = \{(IS, Mirr(up/down/left/right, 1/0))\}$.

Since most graphic transformations use transformation τ multiple times for the initial graphic IS , the connection between multiple transformations is represented by “ \oplus ,” and expressed as $V = \{(IS, \tau_1) \oplus (IS, \tau_2)\}$.

If the initial shape IS overlapped with the new shape S_τ generated after V , the overlapped shape is called emergent shape S_E ($S_E = \{s_{e1}, s_{e2}, \dots, s_{en}\}$), that is, $S_E = IS \cap S_\tau$. If S_E is also transformed, the expression is

$$V = \left\{ \left(is_1, Tran(d_x, d_y, 1) Rota(\pm\theta, 1) Scal(s_x, s_y, 1) \right) \oplus \left(s_{e1}, Tran(d_{ex}, d_{ey}, 1) Rota(\pm\theta_e, 1) Scal(s_{ex}, s_{ey}, 1) \right) \right\}$$

The illustration is shown in Figure 3.

V of the initial shape hexagon is represented by Equation (6):

$$V = \{(is_1, Tran(3, 0, 1)) \oplus (s_{e1}, Rota(+90^\circ, 0))\}. \quad (6)$$

To eliminate overlap between shapes, define $IF IS, S\tau \in Shape$, then $S_T = Shape - S_E$, where S_T represents the terminal shape, expressed as $V = \{(is_1, Tran(d_x, d_y, 1/0) Rota(\theta, 1/0) Scal(s_x, s_y, 1/0), Eli)\}$; where; $Eli = \{eli_1, eli_2, \dots, eli_n\}$; N represents eliminating the overlap of several transforms. An example illustration is shown in Figure 4.

The transformation process of Figure 4 is shown in Equation (7).

$$V = \left\{ (is_1, Rota(-15^\circ, 0)) \oplus \left(is_1, Tran(1 + 3/4, 0, 0) \right) \right\}. \quad (7)$$

If the shape rule is applied to the graph transformed by τ transformation, the expression is defined as $V = \{(IS, \tau_1) \Rightarrow (IS_{nm}, \tau_2)\}$.

Where; $IS_{nm} = \{is_{n1}, is_{n2}, \dots, is_{nm}\}$; n represents the initial shape and m represents the graph after several transformations.

4. APPLICATION EXAMPLE AND OPTIMIZATION DESIGN

Miao batik is mainly distributed in Sichuan, Guizhou, and Yunnan Provinces of China and has a long history. They have numerous design themes and rich patterns, mainly including natural forms such as flowers, leaves, birds, and fish. Batik products include tablecloths, clothing, aprons, cloth bags, and other daily necessities [17].

Taking Miao batik pattern elements as an example, the predicate shape grammar is used to reuse the patterns, and PSO is used to optimize the number of patterns to determine the Euclidean transformation parameters, so as to determine the Euclidean transformation parameters, so that the patterns can be transformed and reused according to the optimized parameters.

4.1. Batik Pattern Extraction

This paper selects the pattern elements of batik from the website of “Intangible Cultural Heritage of Guizhou Province” as the basis of reuse design. Website system as shown in Figure 5.

Batik patterns are selected from the batik productions example library, as shown in Figure 6.

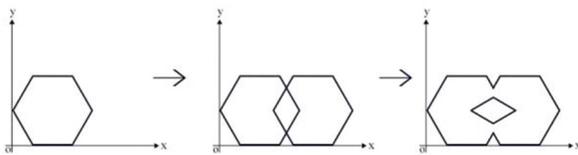


Figure 3 | S_E transform graphic demonstration.

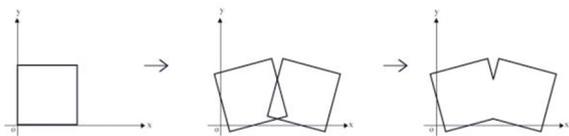


Figure 4 | Eliminate overlapping graphic presentations demonstration.

The pattern element identified by the red box in Figure 6 was selected as the extraction object. When extracting, each pattern element was regarded as a separate closed contour, and the shape context method is used to describe and match the pattern elements [18,19]. The Canny operator [20] is also used to obtain the contour of the shape of the pattern by sampling to obtain the contour sampling points, and then plane software was used to modify the extracted patterns. The results are shown in Figure 7.

4.2. Application of Improved Shape Grammar

Batik patterns include the main, corner, and border pattern elements [21], with special attention to the pattern element arrangement symmetry, single pattern irregularity, and the overall



Figure 5 | System home.

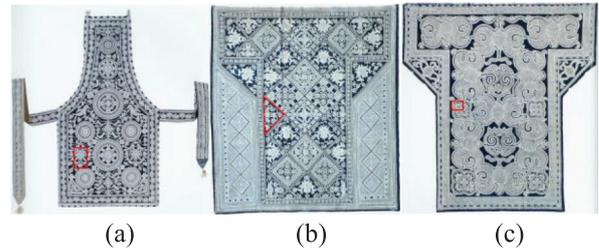


Figure 6 | Batik products. (a) Batik product I; (b) Batik product II; (c) Batik product III.

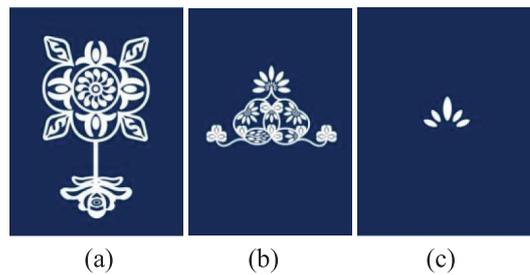
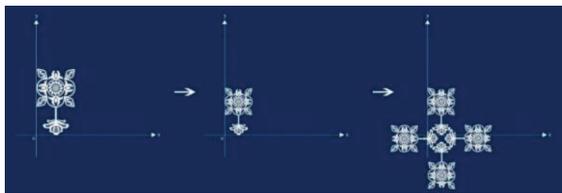


Figure 7 | Pattern matching and extraction results. (a) Patterns extracted from batik product I; (b) Patterns extracted from batik product II; (c) Patterns extracted from batik product III.

spatial arrangement of strong features. Based on these characteristics, it is feasible to characterize the reuse process of pattern elements with improved shape grammar.

Taking the pattern elements are shown in Figure 7(a) and (b) as examples for reuse design. The Figure 7(a) pattern element is set as $pattern_1$, and the Figure 7(b) pattern element is set as $pattern_2$. First, the predicate shape grammar vocabulary $V_{pattern_1}$ and $V_{pattern_2}$ are defined, as shown in Equations (8) and (9), and then the pattern elements of Figure 7(a) and (b) are transformed according to the rules of $V_{pattern_1}$ and $V_{pattern_2}$. The result is shown in Figure 8.

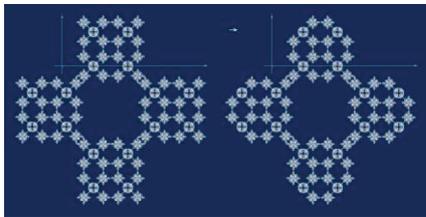
$$V_{pattern_1} = \left\{ \begin{array}{l} V_1 = \left\{ \begin{array}{l} (pattern_1, Scal(1/\sqrt{2}, 1/\sqrt{2}, 0)) \\ \Rightarrow (pattern_{11}, (Circ(1, -0.5), 4)) \end{array} \right\} \\ V_2 = \left\{ \begin{array}{l} (pattern_{12}, Tran(-4, 0, 1)) \\ \Rightarrow (pattern_{13}, (Circ(1, 0), 4)) \end{array} \right\} \\ V_3 = \left\{ \begin{array}{l} (pattern_{13}, Tran(14, 0, 1)) \\ \Rightarrow (pattern_{14}, Mirr(down, 1)) \\ \Rightarrow (pattern_{15}, Rota(+45^\circ, 0)) \end{array} \right\} \end{array} \right\}. \quad (8)$$



(a)



(b)



(c)



(d)

Figure 8 | Predicate shape grammar instance application. (a) Pattern formed according to V_1 rules; (b) Pattern formed according to V_2, V_3 rules; (c) Deletion processing; (d) Pattern formed according to $V_{pattern_2}$ rules.

$$V_{pattern_2} = \left\{ \begin{array}{l} (pattern_2, Scal(3/5, 3/5, 0)) \\ \Rightarrow (pattern_{21}, Mirr(down, 1) Tran(3, 0, 1)) \end{array} \right\}. \quad (9)$$

Similarly, the reused graphics can be added or deleted as required by the design, as shown in Figure 8(c). If the final patterns shown in Figure 8(c) and (d) are subjected to linear array rule processing and then to combination processing, the two patterns can be reused to design articles such as simple tablecloths or wall decorations, as shown in Figure 9.

4.3. Application of Optimization Algorithm

According to the type of batik pattern and the characteristics of the pattern elements in Figure 7(a)–(c), the pattern of Figure 7(a) is used as the main pattern, and Figure 7(b) and (c) are used as the border element pattern. As shown in Figure 10, the circular configuration mode of the batik pattern is selected. The corresponding position is selected according to the different functions of each pattern, and corner elements are not arranged for circular configuration.

When Figure 7(a) ($pattern_1$) is used as the main pattern element, two transformations are defined and as shown in Equations (10) and (11):

$$V_{1_{pattern_1}} = \{(pattern_1, (Circ(0, 0), N))\}. \quad (10)$$

$$V_{2_{pattern_1}} = \left\{ \begin{array}{l} (pattern_1, Scal(1/2, 1/2, 0)) \\ \Rightarrow (pattern_{11}, (Circ(x_0, y_0), 8)) \\ \Rightarrow (pattern_{12}, (Circ(0, 0), N)) \end{array} \right\}. \quad (11)$$

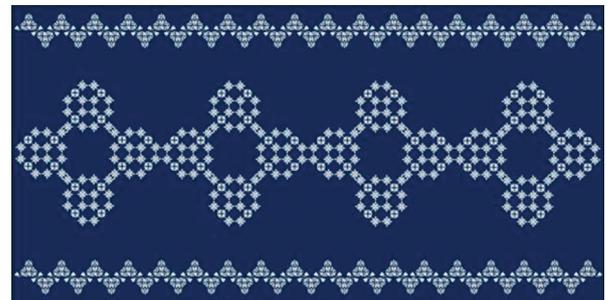


Figure 9 | Reuse design result.

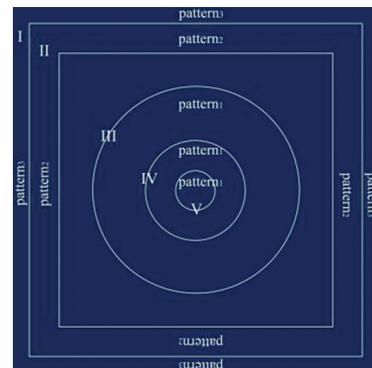


Figure 10 | Configuration and regionalization.

When Figure 6(b) and (c) are used as border pattern elements, two transformation modes were defined, as shown in Equations (12–14), respectively:

$$V_{I_{pattern_2}} = \{(pattern_2, Lnar(d_x, n_x))\}. \quad (12)$$

$$V_{II_{pattern_2}} = \left\{ \begin{array}{l} (pattern_2, Lnar(d_x, n_x)) \\ \Rightarrow (pattern_{21}, Scal(1/2, 1/2, 0)) \\ \Rightarrow (pattern_{22}, Mirr(up/down, 1)) \end{array} \right\}. \quad (13)$$

$$V_{I_{pattern_3}} = \{(pattern_3, Lnar(d_x, n_x))\}. \quad (14)$$

4.3.1. Mathematical model construction

In this study, the PSO algorithm is used to optimize the number of layouts of each pattern at a specified position. First, according to the size parameters shown in Figure 11, the overall mathematical model is established, as shown in Equation (15):

$$\left\{ \begin{array}{l} F(X_n) = f(x_1), f(x_2), \dots, f(x_n) \\ K \cdot x_1 = X \\ F(X_n) \geq 0 \\ \sum K_u F(x_n) = F(X) \end{array} \right. \quad (15)$$

Where; $n = (1, 2, \dots, 5)$, $m = (1, 2, \dots, 5)$, K represents the original number, and u represents the number of variables.

According to the placement position of each pattern in Figure 11 and the size parameters of each part in Figure 11, a mathematical model was established according to the transformation mode of

Equations (10) and (12–14) as the basis of reuse transformation, as shown in Equation (16):

$$\left\{ \begin{array}{l} 15 \leq x_1 \leq 37 \\ 4 \leq x_2 \leq 7 \\ 11 \leq x_3 \leq 24 \\ 5 \leq x_4 \leq 12 \\ 2 \leq x_5 \leq 5 \\ F(X_n) = 2x_1 + 6x_2 + 6x_3 + 6x_4 + 6x_5 \end{array} \right. \quad (16)$$

Where; x_1, \dots, x_5 is a design variable representing the number of rows of fill patterns in each of the five portions in Figure 10(a), $F(X_n) = 2x_1 + 6x_2 + 6x_3 + 6x_4 + 6x_5$ is the objective function, and $15 \leq x_1 \leq 37, \dots, 2 \leq x_5 \leq 5$ are the constraints.

4.3.2. Result analysis

MATLAB R2017 software is used as the optimization tool, the local and global acceleration constants of the PSO were set as $c_1 = c_2 = 0.149445$, and the population size was $S = 200$. The pattern formed by the transformation rule of Equation (10) is used as the main element, and the optimization data are shown in Table 2.

Since x_1, \dots, x_5 represent the number of layouts, rounding is performed. After the five sets of algorithm data generated in Table 2 are rounded, the transformations are arranged in accordance with Figure 10 and Equations (10) and (12–14). If there is a scaling factor for any patterns, the number of additions or subtractions of the pattern is inversely proportional to the scaling factor.

The first set of optimized data was combined with the predicate shape grammar vocabulary V , and the undefined transformation parameters in V are defined according to the optimized number of patterns and the defined scaling ratio. The expression V_G is defined according to the five sets of data in Table 2, thereby generating a pattern, as shown in Figure 12(a)–(e), respectively. The first set of data was used and expression V_{G1} is defined, as shown in Equation (17). The rest of the expressions are similar.

$$V_{G1} = \left\{ \begin{array}{l} V_{I_{pattern_3}} = \{(pattern_3, Lnar(3.65, 20))\} \\ V_{II_{pattern_2}} = \left\{ \begin{array}{l} (pattern_2, Lnar(12, 5)) \Rightarrow \\ (pattern_{21}, Scal(1/2, 1/2, 0)) \Rightarrow \\ (pattern_{22}, Mirr(down, 1) Tran(-3, 0, 1)) \end{array} \right\} \\ V_{III_{pattern_1}} = \{(pattern_1, (Circ(0, 0), 22))\} \\ V_{IV_{pattern_1}} = \left\{ \begin{array}{l} (pattern_1, Scal(1/2, 1/2, 0)) \Rightarrow \\ (pattern_{11}, (Circ(0, 0), 14)) \end{array} \right\} \\ V_{V_{pattern_1}} = \left\{ \begin{array}{l} (pattern_1, Scal(1/3, 1/3, 0)) \Rightarrow \\ (pattern_{11}, (Circ(0, 0), 6)) \end{array} \right\} \end{array} \right. \quad (17)$$

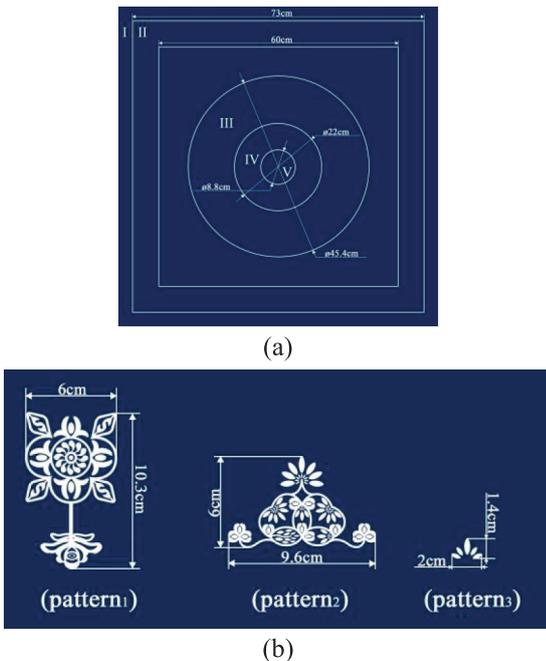


Figure 11 | Dimensional parameters. (a) Each frame size parameters; (b) Initial size parameters of each patterns.

If the pattern formed by the transformation rule of Equation (11) is used as the main element, the constraints x_3, x_4, x_5 were changed,

Table 2 | Selecting five sets of PSO data optimized according to Equation (16).

| Number of Groups \ Variable | x_1 | x_2 | x_3 | x_4 | x_5 |
|-----------------------------|--------|-------|--------|--------|-------|
| Group 1 | 20.431 | 5.181 | 22.288 | 7.360 | 2.060 |
| Group 2 | 33.271 | 6.269 | 18.184 | 7.216 | 4.425 |
| Group 3 | 26.166 | 4.235 | 13.233 | 11.985 | 3.063 |
| Group 4 | 15.326 | 5.236 | 16.459 | 9.425 | 4.383 |
| Group 5 | 18.415 | 6.806 | 23.775 | 6.347 | 4.977 |

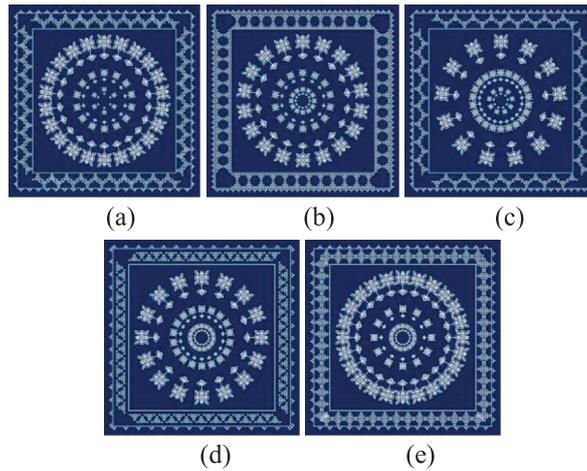


Figure 12 | Pattern element reuse design result I. (a) Group 1 data pattern; (b) Group 2 data pattern; (c) Group 3 data pattern; (d) Group 4 data pattern; (e) Group 5 data pattern.

Table 3 | Selecting five sets of PSO data optimized according to Equation (18).

| Number of Groups \ Variable | x_1 | x_2 | x_3 | x_4 | x_5 |
|-----------------------------|--------|-------|--------|-------|-------|
| Group 1 | 19.785 | 4.881 | 9.713 | 4.231 | 1.985 |
| Group 2 | 32.871 | 6.417 | 6.845 | 5.437 | 1.667 |
| Group 3 | 26.166 | 4.235 | 8.143 | 3.415 | 2.015 |
| Group 4 | 15.326 | 5.236 | 11.306 | 5.843 | 1.788 |
| Group 5 | 17.935 | 6.746 | 9.451 | 3.472 | 2.855 |

as shown in Equation (18). The final optimization data is shown in Table 3, and the design pattern reuse results are shown in Figure 13.

$$\begin{cases} 15 \leq x_1 \leq 37 \\ 4 \leq x_2 \leq 7 \\ 11 \leq x_3 \leq 24 \\ 5 \leq x_4 \leq 12 \\ 2 \leq x_5 \leq 5 \\ F(X_n) = 2x_1 + 6x_2 + 6x_3 + 6x_4 + 6x_5 \end{cases} \quad (18)$$

As seen from that above reuse results, the same transformation rules were applied to the body patterns in Figures 12 and 13, respectively. However, for the border pattern $pattern_2$, different transformation rules are adopted to observe the influence of different border configurations on the pattern and to explore the transformation rules of border styles.

4.4. Construction of Pattern Reuse Design System

In order to apply the predefined improved shape grammar to the reconfiguration design of different pattern frames quickly, the predicate shape grammar expression is applied to the process implementation, and the batik pattern reuse design system is constructed to achieve the rapid pattern frame reuse design. The system implementation is shown in Figure 14.

The system is divided into three main areas: read, operation, and display. The reading area is used to display the pattern elements to be arranged, the operation area is used to input relevant parameters to arrange the pattern elements, and the display area is used to display the results of the operation. The system can obtain the coordinates of the corresponding position by clicking the mouse on the canvas of the display area.

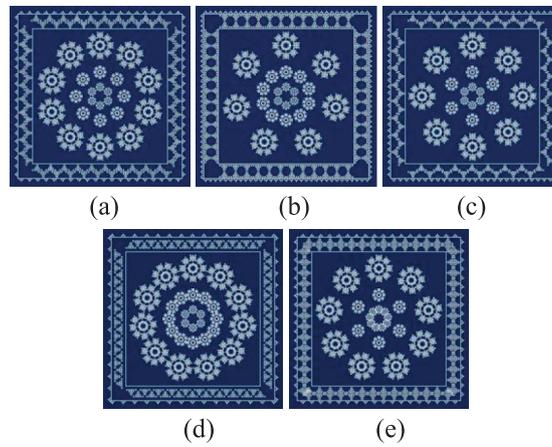


Figure 13 | Pattern element reuse design result II.
 (a) Group 1 data pattern; (b) Group 2 data pattern;
 (c) Group 3 data pattern; (d) Group 4 data pattern; (e)
 Group 5 data pattern.

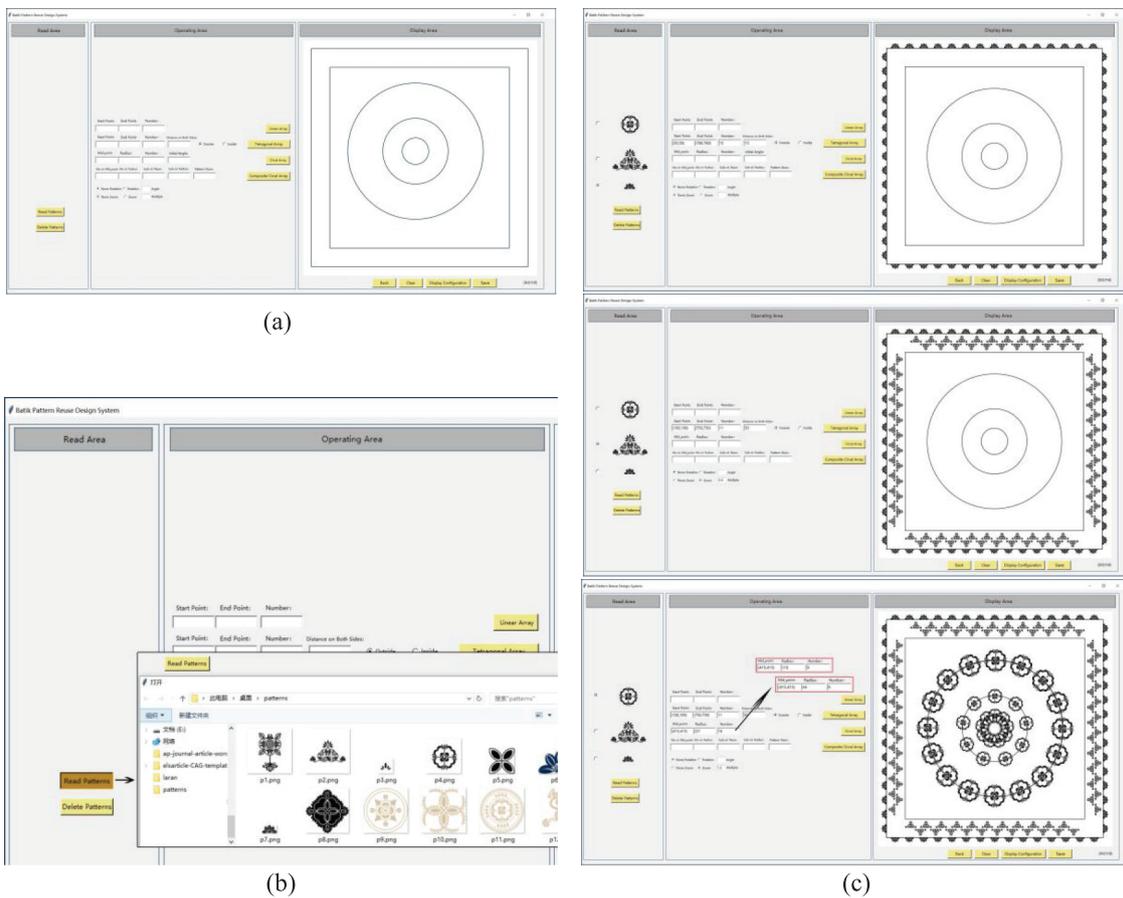


Figure 14 | System design. (a) Initial interface; (b) Reading batik patterns; (c) Apply rules to generate patterns.

5. EXPERIMENTAL COMPARISON AND RESULT ANALYSIS

In this study, the Eyseso eye tracker is used to perform eye movement experiments [22,23] on 10 patterns based on the PSO-optimized data reuse design. Twenty postgraduates are selected and their uncorrected or corrected visual acuity is above 1.0. The subjects

were required to sit 0.5 m in front of the screen and adjust the angle of the eye mover to 15°, as shown in Figure 15.

In the experiment, 10 patterns are arranged out of order into three groups to counteract the influence of the different display position of each pattern on the experimental results. The sample code and arrangement of the experimental group are shown in Figure 16. The experiment measured the fixation time of the subjects on different



Figure 15 | Eye movement experiment scene.

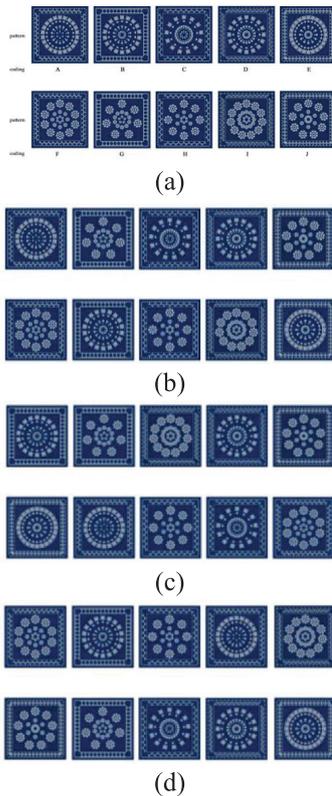
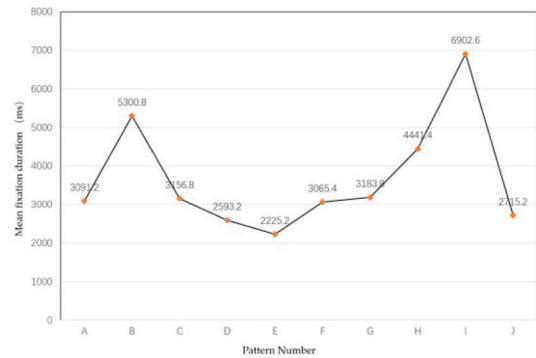


Figure 16 | Sample coding and disordered arrangement of the experimental group. (a) Sample coding; (b) Group 1; (c) Group 2; (d) Group 3.

patterns and calculated the average fixation time of each pattern, as well as the hot spot map of the longest fixation pattern. The experimental results are shown in Figure 17.

The results showed that the mean fixation time of different patterns is in the order of $I > B > H > G > C > A > F > J > D$.



(a)



(b)

Figure 17 | Eye movement test results. (a) Mean fixation time comparison chart; (b) The hotspot map for the longest gaze pattern.

Thus, the subjects pay more attention to the patterns formed from the fourth group of data in Table 3, and the hot spot map shows that the subjects pay more attention to the main pattern elements than the border elements.

The A–E encoded pattern (i.e., the subject pattern formed by the transformation rule of Equation (10)) had a total fixation duration average of 16,357.2 ms. The F–J encoded pattern (i.e., the subject pattern formed by the transformation rule of Equation (11)) had a mean total fixation duration of 20,308.4 ms, indicating that the subjects were more interested in the subject pattern formed by the transformation rule of Equation (11).

6. CONCLUSION AND PERSPECTIVES

1. Aiming at the problem of the low abstraction degree of traditional shape grammar in design pattern reuse, the concept of predicate shape grammar vocabulary is proposed to represent the concrete process of pattern reuse design, which transfers the traditional shape grammar from the initial shape to a transformation process and enhances the abstraction of the shape grammar.
2. The improved shape grammar is applied to the reuse design of Miao batik pattern elements. The results show that the improved shape grammar can satisfy the reuse design of batik pattern elements. It is the first time that improved shape grammar and PSO have been combined to form multiple batik patterns, which proves the good scalability of the method.

3. Through eye movement experiments, it is concluded that the subjects are more interested in the main pattern elements formed by the transformation rules of Equation (11), which shows that complex patterns and compact arrangement can attract people's attention. As for the main pattern formed by the transformation rules of Equations (10) and (11), the pattern formed by the fourth group of data in Table 3 is more noticeable to the subjects according to the second group of Table 2. According to the hot spot map, the subjects pay more attention to the main pattern. Therefore, in the future work, the design emphasis should be placed on the design and coding of the main pattern elements.
4. Future work will focus on deepening the content of Section 4.4, that is, expanding the function of the system for pattern reuse design and improving the flexibility and accuracy of pattern reuse design.

CONFLICT OF INTEREST

The authors declare no conflicts of interest.

AUTHORS' CONTRIBUTIONS

Supervision, N.D.; conceptualization, N.D. and J.L.; research methodology, N.D. and L.H.; experimental operation, N.D. and L.H.; writing—review and editing, N.D. supervision, J.L.; project administration, J.L.

Funding Statement

This research was supported by the Natural Science Foundation of China (Nos. 51865004, 2014BAH05F01) and the Provincial Project Foundation of Guizhou, China (Nos. [2018]1049, [2016]7467).

ACKNOWLEDGMENTS

The authors thank the anonymous reviewers for their valuable suggestions and important comments which greatly helped them to improve the presentation of this manuscript.

REFERENCES

- [1] F. Santos, K. Kwiecinski, A. Almeida, *et al.*, Alternative shaper: a model for automatic design generation, *Form. Asp. Comput.* 30 (2018), 333–349.
- [2] K. Browne, C. McMahon, J. Williams, Describing process plans as the formal semantics of a language of shape, *Artif. Intell. Eng.* 10 (1996), 153–169.
- [3] T. Grasl, A. Economou, From topologies to shapes: parametric shape grammars implemented by graphs, *Environ. Plan. B: Plan. Design.* 40 (2013), 905–922.
- [4] T. Trescak, I. Rodriguez, M. Esteva, General shape grammar interpreter for intelligent designs generations, in *Conference: Sixth International Conference on Computer Graphics, Imaging and Visualization: New Advances and Trends (CGIV 2009)*, Tianjin, China, 2009.
- [5] I. Jowers, C.F. Earl, Implementation of curved shape grammars, *Environ. Plan. B: Plan. Design.* 38 (2011), 616–635.
- [6] T. Strobbe, P. Pauwels, R. Verstraeten, *et al.*, Toward a visual approach in the exploration of shape grammars, *Artif. Intell. Eng. Design. Anal. Manuf.* 29 (2015), 503–512.
- [7] H.Y. Keles, Embedding parts in shape grammars using a parallel particle swarm optimization method on graphics processing units, *Artif. Intell. Eng. Design. Anal. Manuf.* 32 (2018), 256–268.
- [8] H. Chau, A. McKay, C. Earl, *et al.*, Exploiting lattice structures in shape grammar implementations, *Artif. Intell. Eng. Design. Anal. Manuf.* 32 (2018), 147–161.
- [9] C. Jia, T. Ming, Representing 3D shape grammars in a generative product design system, in *Conference: Fifth International Conference on Design Computing and Cognition*, College Station, TX, USA, 2012, pp. 377–392.
- [10] N. Abdelbar, B. Rachid, Parametric shape grammar formalism for Moorish geometric design analysis and generation, *ACM J. Comput. Cultur. Herit.* 10 (2017), 19–39.
- [11] C. Jia, T. Ming, Integrating shape grammars into a generative system for Zhuang ethnic embroidery design exploration, *Comput. Aided Design.* 45 (2013), 591–604.
- [12] S. Zahra, U. Hassan, I.J. Palmer, Parameterized shape grammar for generating n-fold Islamic geometric motifs, in *Conference: Cyberworlds 2015*, Visby, Sweden, 2015.
- [13] G. Stiny, J. Gips, Shape grammars and the generative specification of painting and sculpture, in: C.V. Freeman (Ed.), *Information Processing*, North-Holland, Amsterdam, Netherlands, 1972, pp. 1460–1465.
- [14] G. Stiny, Introduction to shape schema grammars, *Environ. Plan. B: Plan. Design.* 7 (1980), 343–351.
- [15] R. Poli, Analysis of the publications on the applications of particle swarm optimizations, *J. Artif. Evol. Appl.* 2008 (2008), 1–10.
- [16] X. Fei, L. Xiao, C. Hui, H. Hui, Application of particle swarm optimization to the estimation of the TSInSAR deformation parameter, *Remote Sens. Lett.* 10 (2019), 756–765.
- [17] B. Xiao, The psychological study of ethnic batik pattern composition based on social psychology: take the Guizhou Miao, Buyi nationality as an example, *Guizhou Ethnic Stud.* 36 (2015), 79–82.
- [18] S.B.J.M. Puzicha, Shape context: a new descriptor for shape matching and object recognition, *Pattern Anal. Mach. Intell.* 24 (2002), 509–522.
- [19] J. Geng, Z. Hai, Z. Qian, L. Xue, Symmetry invariance shape context feature extraction and image matching, *J. Graph.* 39 (2018), 470–476.
- [20] H.R. Shahdoosti, Z. Rahemi, Edge-preserving image denoising using a deep convolutional neural network, *Signal Process.* 159 (2019), 20–32.
- [21] Y. Hai, P. Hong, Y. Yi, *et al.*, A generation method of carpet pattern based on topology configuration, *J. Comput. Aided Design Comput. Graph.* 25 (2013), 502–509.
- [22] W. Shi, S. Xiao, H. Yi, *et al.*, Distributed attentive user interface for cross-device interaction based on eye tracking, *J. Comput. Aided Design Comput. Graph.* 29 (2017), 1713–1724.
- [23] W. Chen, G. Bin, W. Lin, L. Yi, Extraction and application research of culture design gene based on eye movement analysis, *Comput. Eng. Appl.* 54 (2018), 217–224, 235.