Computer-Aided Design 128 (2020) 102911

Contents lists available at ScienceDirect

Computer-Aided Design

journal homepage: www.elsevier.com/locate/cad

An Efficient Pattern Design Method for Plush Toys Using Component-Based Templates[☆]

Yinghan Jin[a], Wanting Feng[a], Long Zeng[b], Dongliang Zhang[a,*]

[a] *College of Computer Science, Zhejiang University, Hangzhou, 310058, China*
[b] *Graduated School at Shenzhen, Tsinghua University, Shenzhen, 518055, China*

ARTICLE INFO

Article history:
Received 10 January 2020
Received in revised form 16 June 2020
Accepted 22 June 2020

Keywords:
Pattern design
Component-based modeling
Design template
Plush toy design

ABSTRACT

In pattern design of a 3D plush toy model, the key task is to draw seam lines on the 3D surface to divide it into surface patches to obtain 2D patterns. However, how to draw seam lines requires a lot of professional knowledge and skills of pattern design. In this paper, we present a new method that can guide and assist users to create 2D patterns of a 3D toy model efficiently. Based on a component-based modeling method, we build a template library including the information of 3D components and seam lines. To design patterns of a toy, firstly, the most similar template will be retrieved from the library, then seam lines will be generated on the 3D surface automatically by mapping the template. Finally, 2D patterns will be obtained using a flattening method. The proposed method is easy-to-lean and easy-to-use, and it does not require the user to have much experience of pattern design. It can improve the efficiency of design and shorten the learning time for new designers and non-professionals.

1. Introduction

In the industry of plush toys, the production process of a toy includes pattern design, fabric cutting and sewing, etc. Among them, pattern design is to obtain 2D patterns of a toy, which is an essential step of toy design. For a long time, pattern design has been completed by traditional manual work, which means toy designers create 2D patterns manually based on their spatial imagination, practical experience and aesthetic ability. In practice, due to the lack of design experience and ability, sometimes the patterns designed by a designer cannot meet the quality requirements and need to be modified several times, which wastes materials, labor and time.

With the development of computer graphics and computer-aided design techniques, pattern design of toys can be realized by 3D design methods. The 3D technology improves the efficiency and accuracy of design, and alleviates the requirement of space imagination for designers. However, learning the skills of pattern design remains a difficult problem. As we know, pattern design is a traditional and professional craft, the design knowledge is non-formulaic and non-standard, and it is difficult to be expressed clearly and taught directly. Normally, learning design mainly depends on long-term teaching by teachers. Whether learners can master it largely depends on their comprehensive ability. This greatly raises the threshold of using the technique of 3D toy design.

To design patterns of a 3D toy model, firstly the designer needs to analyze the shape and structure of the model, then design and draw appropriate seam lines, and finally achieve 2D patterns by flattening 3D surface patches. In the process, the main knowledge and skills difficult to be mastered are drawing seam lines, which are curved lines drawn on the surface of a 3D model to divide the surface into several closed regions. Designing seam lines is the key step of pattern design for 3D toy models. It needs professional knowledge, and the way of drawing seam lines is related to the overall shape of the model, the anatomical structure of different parts, the functional and aesthetic requirements of the toy.

In the practice of pattern design, toy parts with similar shapes usually have similar designs of seam lines. Therefore, one design of a toy can be applied to another toy. The reuse of design can greatly improve the efficiency of design. Inspired by the idea of design reuse, we propose a novel method of designing patterns for 3D toy models using component-based templates. In the approach, we use component-based 3D toy models, and utilize templates to store the information of professionally designed seam lines of classified toy components. To design patterns of a toy, the most similar component will be retrieved, and the corresponding seam lines will be mapped automatically to the surface of the toy model, and finally the patterns are gotten by flattening 3D surface patches. The reuse of design knowledge is achieved by the component-based template. Thus, the automation of pattern design can be realized, the efficiency of design can be improved, and the difficulty of learning can be reduced.

☆ This paper has been recommended for acceptance by Charlie Wang.
* Corresponding author.
E-mail address: dzhang@zju.edu.cn (D. Zhang).

https://doi.org/10.1016/j.cad.2020.102911
0010-4485/

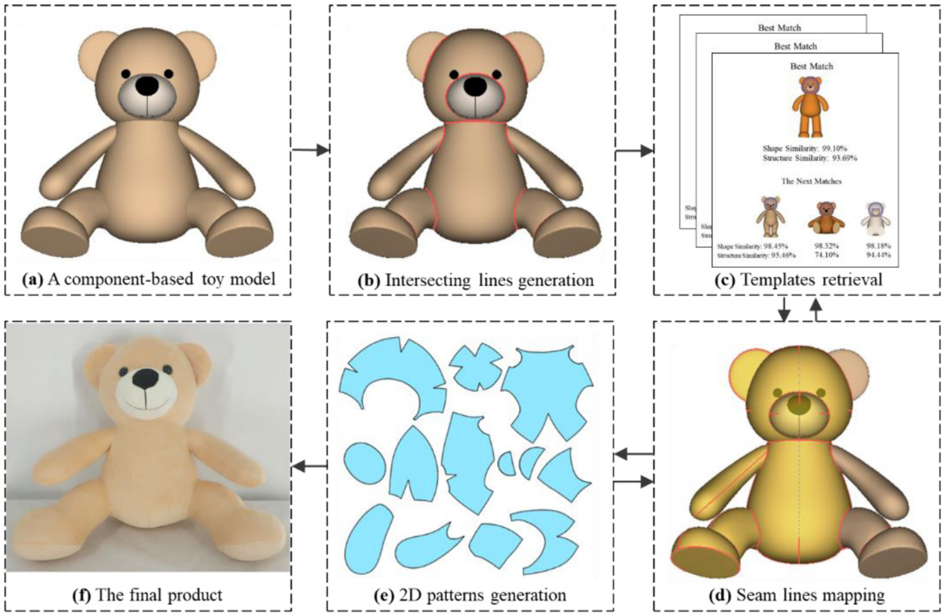


Fig. 1. The illustration of our method.

Fig. 1 shows a brief illustration of our method using a bear model as an example. Firstly, the intersecting lines among different components are generated automatically (Fig. 1b). Then, the most similar templates will be searched from the template library for each component (Fig. 1c). The user can choose the most suitable template to map the seam lines on the surface of the component (Fig. 1d). Seam lines can be further refined interactively until the user is satisfied with the generated 2D patterns of the toy (Fig. 1e). Finally, a physical toy is manufactured based on the designed patterns, as shown in Fig. 1f.

The following sections of the paper are organized as follows: Section 2 reviews related works on component-based modeling, pattern design of soft products and 3D model retrieval. Section 3 introduces the component-based toy design method, and Sections 4–6 elaborate on several steps of the proposed method. Experimental results are given in Section 7. At last, we summarize and discuss the contributions, limitations and future work of the paper in Section 8.

2. Related work

Component-based 3D modeling: Easy-to-use 3D modeling techniques in CAD systems are widely studied by researchers. Igarashi et al. [1] came up with a sketching modeling system *Teddy* which opened a new area of 3D modeling. The 3D modeling system *EasyToy* proposed by Liu et al. [2] allowed the user to create complicated cartoon models by sketching modeling and assembling components. With rich data of existing models, it is much easier to create a new model by reusing components. Funkhouser et al. [3] put forward a method to generate a complex 3D model by manually cutting and bonding parts from existing shapes. Chaudhuri et al. [4] proposed an assembly-based modeling system using a Bayesian network to predicted relevant components and presenting them to the user. Jain et al. [5] proposed a 3D modeling system to synthesize new models by recombining parts with hierarchy and contact information. Chaudhuri and Koltun [6] presented a data-driven approach to give creativity supports of 3D modeling to users by calculation. Kim et al. [7] solved the problem of extracting parts from unstructured and unlabeled model clusters by using an initial template. Xue et al. [8] proposed an example-based 3D reconstruction algorithm to generate 3D models from the 2D drawing using MAP estimation. Kwok et al. [9] extended the example-based modeling to the fashion design with a system that can generate garment designs automatically. These component-based methods make the design work simple and free the designers from trivial work, but they are mainly used for 3D modeling. In this paper, we extend the component-based modeling method to develop a component-based template to design patterns for toys.

Pattern design of soft products: Many studies have been conducted for designing patterns of soft products including plush toys and clothes. Mori and Igarashi [10,11] presented an interactive pattern design method for plush toys. The interface is user-friendly so that the system is suitable for non-professionals to design relatively simple plush toys. Skouras et al. [12] provided an interactive tool for designing inflatable structures. In their approach, physics-based pattern optimization, coarse-scale simulation, and constraint-optimization are used, and the produced results are accurate. Wang et al. [13] presented a method for precise pattern design using constrained contour curves and style curves, which can be used not only for garments but also for toys and shoes. In [14], they proposed a physical/geometric method to compute planar patterns for compression garments. Nobuyuki et al. [15] presented an intuitive garment design tool enabling interactive bidirectional editing between 2D patterns and 3D high-fidelity simulated draped forms. In these methods, the patterns are mainly obtained by flattening 3D surfaces, and the user needs to draw the seam lines on the 3D model manually, which requires a lot of design experience and skills. In our work, we propose a design template by utilizing existing professional design data to create seam lines on 3D models automatically.

3D model retrieval: 3D model retrieval has wide applications in CG and CAD. The core task of content-based retrieval is defining an appropriate descriptor to represent features and calculating similarity. Paquet et al. [16] proposed a method to describe both 2D and 3D shape information using a bounding box descriptor. Ohbuchi et al. [17] presented a multiresolution shape descriptor based on 3D alpha shapes to achieve 3D shape similarity comparison. To describe 3D shape features simply, Vranic et al. [18] proposed a ray-based method of shape characterization. Liu et al. [19] presented a multi-modal clique graph (MCG) matching method and an image set-based clique/edgewise similarity measure for multi-view matching. Wang et al. [20] proposed a method of retrieving 3D models from 2D sketches. In their method, two siamese convolutional neural networks are adopted. Nie et al. [21] introduced a weakly supervised learning method based on a sparse representation-based classification to retrieve 3D objects. Theologou [22] introduced a part-based descriptor for the case of articulated objects. These methods are only used for retrieving 3D models with similar shapes. In our method, we not only retrieve 3D models with similar shapes, but also similar structures.

3. Component-based 3D toy design

3.1. Component-based toy modeling

In this paper, the 3D toy model for pattern design is built by assembling several 3D components, which means that all parts of a 3D model are independent. Each component is represented by a 3D genus-zero mesh that has a triangular surface. The user can create a toy model freely by sketching 3D components or combining components from a library, which includes some frequently used components of plush toys, such as different kinds of head, body, arm and foot, etc. (Fig. 2a). The component can be transformed, including translation, rotation, scaling, symmetry and deformation, to an appropriate shape and position, then assembled to build a toy model, as shown in Fig. 2b.

Component-based toy modeling is a useful modeling method. It has two main advantages:

(1) **Fast modeling**. On the basis of component modeling, components can be reused so that 3D models can be built quickly. With the help of the component library, the modeling process is simplified and accelerated. If a part of the model to be created can be found in the component library, there is no need to re-build it. As shown in Fig. 3, four bear models with different postures are created using the same components.

(2) **Easy modification.** 3D toy modeling is an iterative procedure that needs many modifications. Assembling several toy components into a whole model will reduce the difficulty of modifications. For example, the position of the horn of a unicorn's head can be easily modified by rotation and translation, as shown in Fig. 4a. If the horn and head are merged into a whole, adjusting the position of the horn needs the operations of cutting, surface recovering and surface extruding, as shown in Fig. 4b. The process of modification is very complex and time-consuming.

Modeling by components is easy-understanding and conforms to the general process of 3D toy modeling. Fig. 5 shows some component-based models which are rebuilt from off-the-shelf toys. The models were created by designers using a sketch-based 3D modeling system EasyToy [2]. In the practice of toy design, even if the final toy model is represented by only one mesh, the process of modeling is usually component-based, which means that the final model is created by merging several parts, which are modeled one by one in the design stage.

From the above, we know that the component-based modeling method is feasible and suitable for 3D toy design. Based on the components, we can not only reuse them for 3D modeling, but also reuse them for pattern design. Our proposed method of pattern design is based on component modeling, in which each component has a closed surface.

If a 3D toy model is formed as a whole model, which does not consist of components, in this case, it is necessary to segment the whole model into several components before using the proposed method. Some researchers [23,24] have conducted research on the automatic segmentation of 3D models using neural networks. It is possible to convert a segmented part into a component with a closed surface. We will conduct such research as one of our future work.

3.2. Component-based pattern design

Based on the component-based toy model, we use component-based templates to design patterns. A template contains the information of the 3D shape and seam lines of a 3D toy component so that it can be reused to apply the design to a new component with a similar shape. To deal with various toy models, a template library will be constructed based on a series of typical toy models. To build a template library, firstly, seam lines are drawn on 3D toy models by professional designers. Then, each component will be parameterized, and the component descriptor including the information of 3D shape and seam lines will be generated and stored into the library. Fig. 6a shows the process of constructing the template library. The details of surface parameterization and computing component descriptor will be described in the following sections.

To design patterns of a component-based model, firstly the intersecting lines among intersecting components will be generated automatically. To design seam lines for a specified 3D component, the component will be parameterized and its component descriptor will be generated, then the most similar template will be retrieved from the template library by computing the similarity of shape and structure between component descriptors. The user can choose one suitable template from several most matched templates, then map its seam lines to the component to be designed. Seam lines can be further modified interactively by the user. At last, the surface patches divided by seam lines will be unwrapped into 2D patterns using surface flattening. These steps can be repeated until the user is satisfied with the designed result. Fig. 6b shows the flowchart of designing patterns using component-based templates.

4. Seam line generation and mapping

Seam lines are very important to design patterns for 3D toy models. They are used to divide the surface of a model into several surface patches, which can be flattened into 2D patterns. Before saving a template, seam lines should be drawn on the surface of a 3D component. To apply a template to a new component, seam lines will be mapped on the target component. The correspondence between two components can be established using surface parameterization.

4.1. Editable seam lines

To draw accurate and smooth seam lines on the surface of a 3D toy model, we employ the cubic spline curve [25], which is generated by a cubic equation through a set of control points. Based on the control points drawn by the user, a cubic spline curve is fitted. Then the spline is projected to the adjacent 3D mesh faces to form a trajectory line on the surface, which is a seam line of the model. Fig. 7a shows seam lines drawn by the user on the surface of a component, and Fig. 7b shows the triangular mesh of the 3D component. When a control point of a seam line is adjusted to a new position, a modified seam line will be generated automatically.

Seam lines are the basis of the division of surface patches. It is necessary to record some information of seam lines, including connections and intersections between them and intersections with the edges of the 3D mesh. This information is essential for the selection of 3D surface patches in the flattening process and boundary generation of 2D patterns after flattening.

4.2. Intersecting lines

As mentioned in Section 3, a toy model consists of several components, so there are some components intersecting with

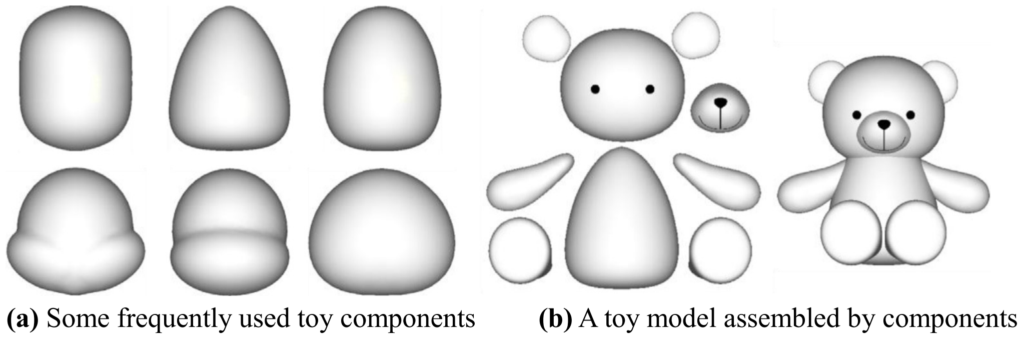


Fig. 2. Component-based 3D toy modeling.

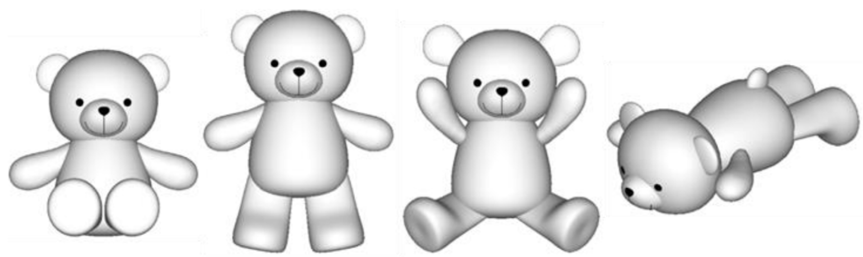


Fig. 3. Different models assembled by same components.

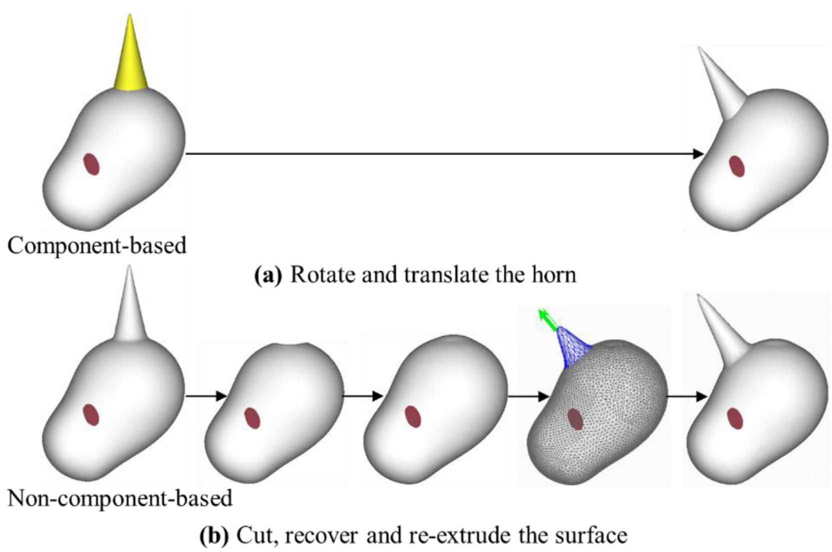


Fig. 4. Component-based modeling simplifies the process of modification.

each other. For two intersecting components, some parts of them are overlapped and invisible, as shown in Fig. 8a. The invisible surfaces should be ignored, otherwise, the surface of a 3D model cannot be divided correctly into several patches and the final 2D patterns cannot meet the requirements of the design. Therefore, it is necessary to create the intersecting line on two overlapping components to separate the visible and invisible surfaces. An intersecting line can be regarded as a special seam line. It can be created automatically by finding intersecting line segments between the triangular meshes of two components and connecting them into a closed-loop. Fig. 8b shows the intersection lines of two overlapped components.

4.3. Surface parameterization

To map seam lines from a template component to a new component, the correspondence between the two components should be established. To do so, we use a surface parameterization method to map the component onto a unit sphere.

The goal of parameterization is to form a mapping relation $f: M_1 \rightarrow M_2$, which maps every seam line of the template model M_1 onto the target model M_2. Since the component is a genus-zero mesh, which has a topologically equivalent embedding on a sphere, the component can be parameterized into a unit sphere. In our approach, we adopt the Alexa algorithm [26] for parameterization. For a triangular 3D model with faces F, edges E, and vertices $V = \{\boldsymbol{v}_0, \boldsymbol{v}_1 \ldots\}$, in each relaxation process, vertex $\boldsymbol{v}_i$ is moved by the following equation:

$$\boldsymbol{p}_i = c\,\frac{\Sigma_{\{i,j\}\in E}\left(\boldsymbol{v}_i - \boldsymbol{v}_j\right)\left\|\boldsymbol{v}_i - \boldsymbol{v}_j\right\|}{\#\{(i,j)|\{i,j\}\in E\}} \tag{1}$$

where c is a constant to control the overall move length. We use the inverse of the longest edge incident upon one vertex to

Fig. 5. Component-based models rebuilt from physical toy products.

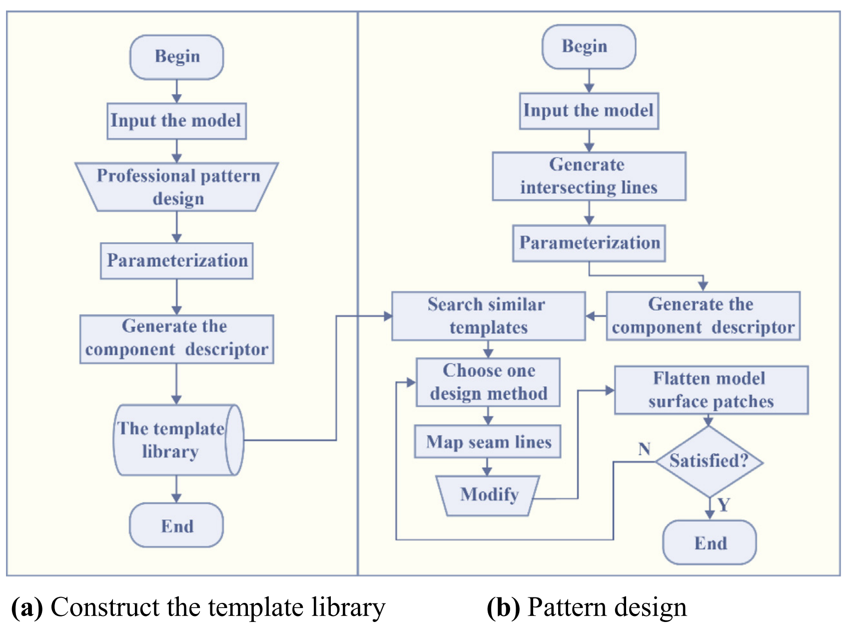


Fig. 6. Flowchart of component-based pattern design.

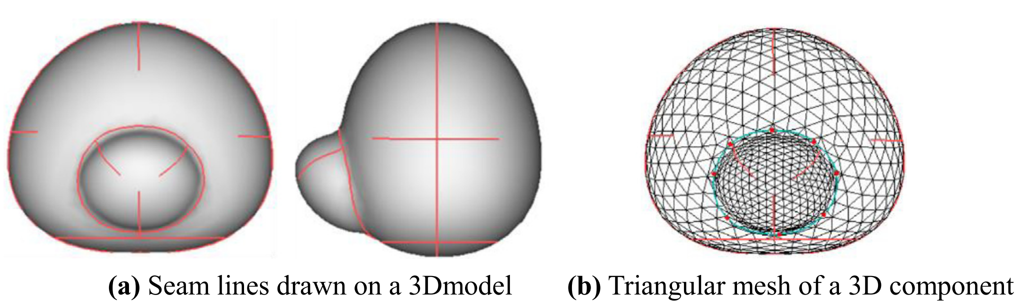


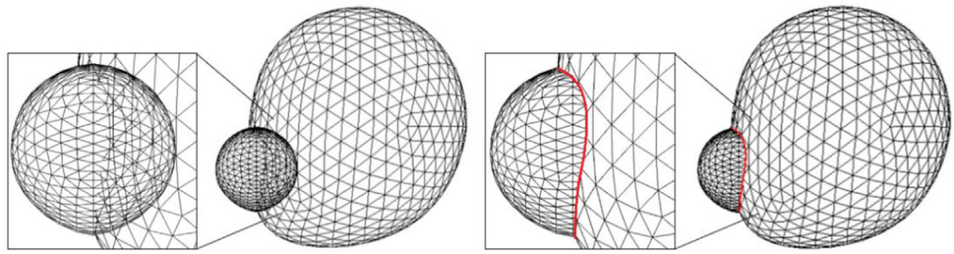
Fig. 7. Draw seam lines on a 3D component.

accelerate the convergence, letting

$$c = \frac{1}{\max(l_E)\,|\{i,j\} \in E} \tag{2}$$

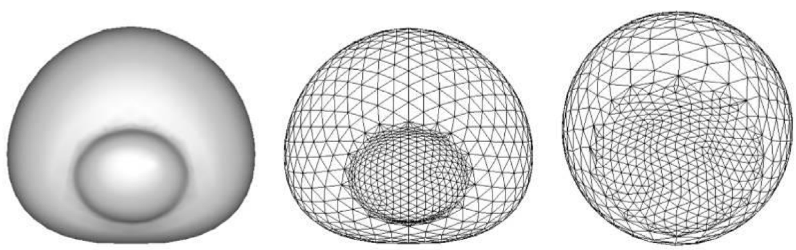
And then make the moved vertex $\boldsymbol{v}_i$ on the unit sphere:

$$\boldsymbol{v}_i^{r+1} = \frac{\boldsymbol{v}_i^r + \boldsymbol{p}_i^r}{\left\| \boldsymbol{v}_i^r + \boldsymbol{p}_i^r \right\|} \tag{3}$$



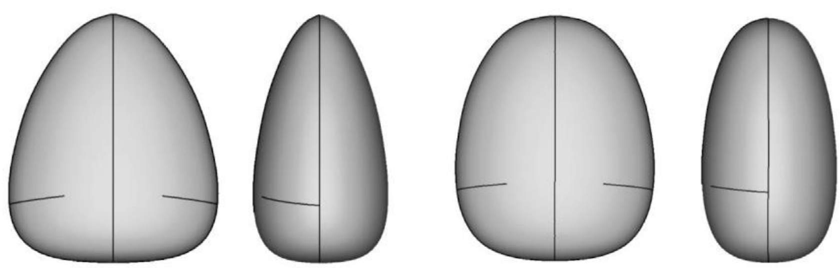
(a) Two intersecting components **(b)** Intersecting lines of two components

Fig. 8. Generate intersecting lines of two overlapped components.



(a) Given model **(b)** Mesh presentation **(c)** Parameterized result

Fig. 9. Parameterization of a 3D component.



(a) A component with seam lines **(b)** Mapping result on another component

Fig. 10. Seam line mapping between two components.

where r is the number of iterations during the relaxation process.

The number of iterations is determined by checking $\mathrm{sgn}((v_0 \times v_1) \cdot v_2)$ in each face. The iteration is considered terminated when three vertices of each face make a clockwise turn. Fig. 9 shows the parameterization result of a 3D component.

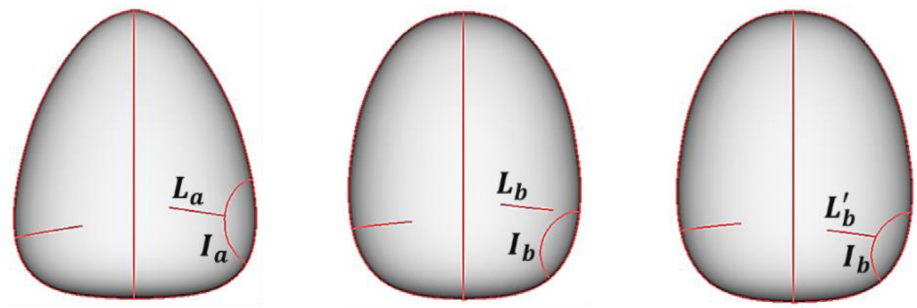
4.4. Seam line mapping

After parameterizing two components M_1 and M_2 into unit spheres S_1 and S_2 respectively, the mapping relation of the two components is established. The next step is to build the mapping relation of seam lines on two parameterized spheres and map them from M_1 to M_2. Since a seam line is formed by a set of control points, we only need to calculate the positions of mapped control points on M_2, and a new seam line can be generated by these points.

Suppose that a control point of component M_1 is P_1, its corresponding point on S_1 is P_1', the mapped control point on component M_2 is P_2, its corresponding point on S_2 is P_2'. For a control point on a triangle of M_1, the barycentric coordinate is used to describe its relative position in the triangle. Thus, the position of the corresponding point on S_1 will be calculated based on its barycentric coordinate. The position of a point on the unit sphere is represented by the polar coordinate. According to the mapping relation, the polar coordinate of a mapped control point on S_1 is the same as that on S_2, which means that P_1' and P_2' have the same polar coordinate. Then, we can find the triangle on S_2 where P_2' is located by searching its nearest triangle. After computing the barycentric coordinate of P_2' related to the triangle, the mapped control point P_2 on M_2 will be calculated. After all control points of a seam line are mapped, a new seam line will be generated on M_2. Fig. 10 shows the result of mapping seam lines from one component to another component.

4.5. Topology maintenance

As a special seam line, the intersecting line is generated automatically based on the target component. It is different from a normal seam line of the target component, which is mapped from the template. Therefore, after applying a template to the target component, the topology of connections among seam lines may be changed. Fig. 11 shows an example of mapping a seam line and an intersecting line. In the figure, L_a is a normal seam line, I_a is an intersecting seam line, and one endpoint of L_a is on I_a. After mapping to the target component, the mapped seam



(a) Seam line template **(b)** Mapped seam lines **(c)** Expected result

Fig. 11. Topology maintenance of seam lines.

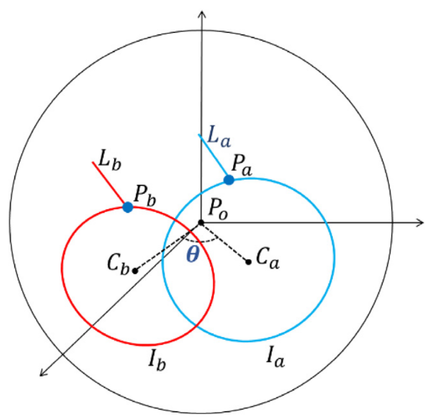


Fig. 12. Adjust seam lines to maintain topology.

line L_b and intersecting line I_b are not connected, as shown in Fig. 11b. In this case, the seam lines cannot divide the surface of the 3D component correctly, which will lead to inappropriate patterns. For a good mapping result, the connection topology of seam lines should be kept and their relative position should be maintained. Fig. 11c shows the expected mapping result of the target component.

In our method, if a seam line is connected to an intersection line of the template component, we will adjust the seam line mapped to the target component to maintain the topology and relative position. Take the case in Fig. 12 for example. In the figure, L_a is a normal seam line, I_a is an intersecting line, the endpoint P_a of L_a is on I_a, I_b is an intersecting line of the target component, and L_b is the expected seam line to be mapped. All lines are shown on a unit sphere. When mapping L_a to the target component, the closest intersecting line I_b will be founded, which will be regarded as the corresponding intersecting line of I_a. The position of L_b will be computed by keeping the topology and relative position between L_b and I_b. To do so, we calculate the rotating angle $\theta = \langle \overrightarrow{C_aP_o}, \overrightarrow{C_bP_o} \rangle$, where C_a and C_b are the centers of the intersecting lines of I_a and I_b respectively, and P_o is the center of the unit sphere, then control points of L_b is gotten by rotating the corresponding points in L_a in the same way as rotating C_a to C_b using P_o as the rotating center. The position of P_b will be further adjusted to make it on I_b by getting the intersecting point between L_b with I_b.

Fig. 13 shows an example of mapping seam lines using maintaining topology. In the figure, the template component and the target component have the same topology, but the shape and position of components are different, so the intersecting lines on the target component are different from those on the template component. Using topology maintenance, reasonable seam lines can be generated.

5. Template retrieval

Template retrieval is to find the most similar template in the library for mapping seam lines to a target component. It includes the following three steps: (1) Model preprocessing, which is to normalize the 3D component. (2) Component descriptor generation, which is to extract an appropriate shape feature vector of each 3D component and save it with seam lines in the template. (3) 3D model retrieval, which is to search the library to find the most matched template for the target component based on component descriptors.

5.1. Model preprocessing

Model preprocessing is used for 3D model retrieval. It normalizes a 3D model to ensure the uniqueness when generating the feature descriptor of a 3D model even if the model is rotated, scaled, or translated. A general approach to realize the model normalization is standardizing the model coordinates using the principal component analysis (PCA). However, sometimes the normalized coordinates of the same model may be different due to the different sizes of models. In our approach, the improved PCA [18] is used to solve this problem. For the vertices of a triangular mesh: $V = \{\boldsymbol{v}_0, \boldsymbol{v}_1 \ldots\}$, we define mean vertex $\overline{\boldsymbol{m}}$ which determined by:

$$\overline{\boldsymbol{m}} = \frac{1}{N} \sum_{k=1}^{N} w_k \boldsymbol{v}_k \tag{4}$$

where $w_k = \frac{N \cdot S_k}{3S}$, $k = 1, 2 \ldots N$. S is the sum area of the triangles of the mesh, and S_k is the sum area of the triangles which have $\boldsymbol{v}_k$ as a vertex.

Then the covariance matrix C is defined by:

$$C = \frac{1}{\sum w_k} \sum_{k=1}^{N} w_k (\boldsymbol{v}_k - \overline{\boldsymbol{m}}) \cdot (\boldsymbol{v}_k - \overline{\boldsymbol{m}})^T \tag{5}$$

Eigenvectors of the matrix C are normalized and sorted in a descending order according to their eigenvalues. After transposing, the transformation matrix A is gotten and then it is used to rotate the vertices:

$$\boldsymbol{v}'_k = A\left(\boldsymbol{v}_k - \overline{\boldsymbol{m}}\right), k = 1, 2 \ldots N \tag{6}$$

where $\mathrm{A} = \left[E^{(0)} \quad E^{(1)} \quad E^{(2)}\right]^T$ *with* $\left\|E^{(n)}\right\| = 1, n = 0, 1, 2.$

Fig. 14 shows different orientations of several components before and after transforming the improved PCA. After preprocessing, the mass center of a component moves to the original of the coordinate system, and the largest variance of the component is distributed along the x-axis, with its second-largest variance occurring along the y-axis.

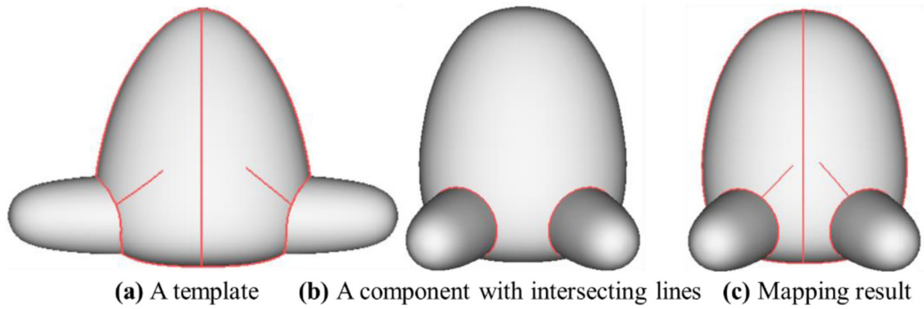


Fig. 13. Mapping seam lines by maintaining topology.

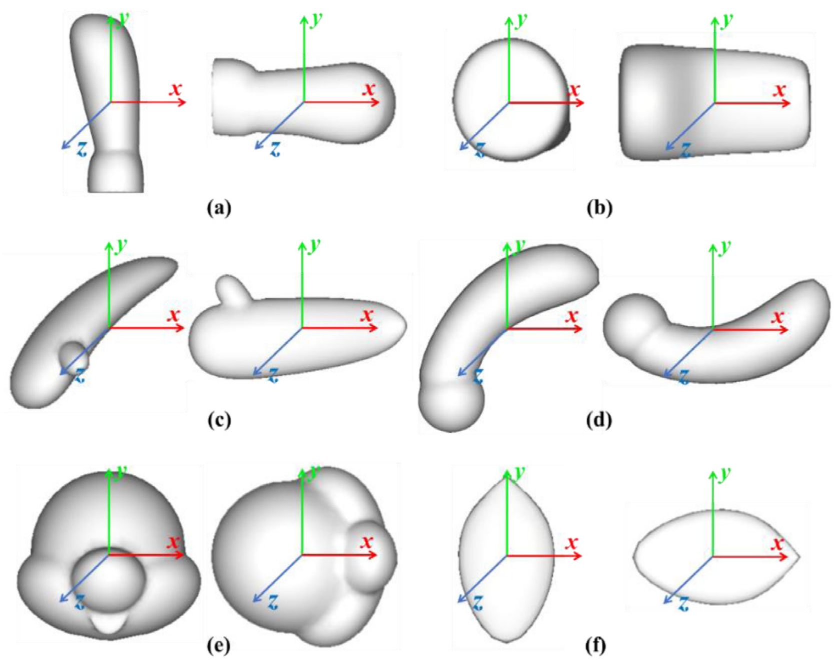


Fig. 14. 3D model normalization.

5.2. Component descriptor generation

The general method of 3D model retrieval is to compute shape descriptors of models and compare the distances between descriptors. In our method, the goal of model retrieval is for mapping seam lines from a suitable template to the target component, the component descriptor not only contains shape features of a component but also the information of seam lines.

In the study, we use the ray-based method [18] to extract shape features of a component. We make use of its advantages of low storage cost, fast speed and reasonably high accuracy for 3D model retrieval. Fig. 15 shows the process of generating a component descriptor. After model preprocessing, the shape features of the component are extracted. For each direction of rays that start from the coordinate origin to twenty vertices of the dodecahedron, we calculate the farthest point intersecting with the normalized component as a feature f_n of the component, and obtain shape features descriptor $\boldsymbol{f}$: $\{f_1, f_2, \dots f_{20}\}$. As mentioned in Section 4.4, to easily represent the information of seam lines of the component and realize the mapping of two components, the seam lines will be mapped onto the unit sphere. Thus, the component will be parameterized to build the mapping relation between it and the unit sphere. After parameterization, the positions of seam lines on the unit sphere can be calculated according to the relation of mapping. In our method, the polar coordinate is used to represent the positions of seam lines, and the information of seam lines on the unit sphere is represented as $\boldsymbol{s}$: $\{s_1$: $Point_1$, $Point_2 \dots Point_i, \dots s_m$: $Point_1$, $Point_2 \dots Point_j\}$. After extracting shape features and calculating the mapped positions of seam lines, a component descriptor consisting of $\boldsymbol{f}$ and $\boldsymbol{s}$ is generated.

5.3. 3D model retrieval

In our method, when we apply a template to the target component, we not only take account of the shape similarity, but also the structure similarity of the component.

Shape similarity. The shape similarity of two components is measured by the distance of their shape features. For a target component, its shape features $\boldsymbol{g}$ will be extracted and the distance between $\boldsymbol{g}$ and each $\boldsymbol{f}$ in the template library will be calculated. The distance between two descriptors is calculated as follows [18]:

$$dist\,(\boldsymbol{f}, \boldsymbol{g}) = 100(1 - \frac{\sum_{i=1}^{20} |f_i - g_i|}{2\sqrt{10}}) \tag{7}$$

where $\|\boldsymbol{f}\| = \|\boldsymbol{g}\| = 1$, and $max_{\boldsymbol{f},\boldsymbol{g} \in R^{20}} \left(\sum_{i=1}^{20} |f_i - g_i|\right) = 2\sqrt{10}$.

Structure similarity. If two components have the same number of intersecting lines and the lines are located at similar positions, we consider that they have similar structures. For a

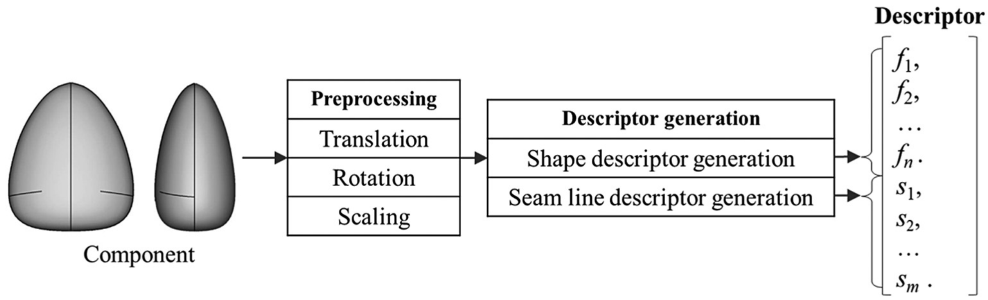


Fig. 15. Process of generating a component descriptor.

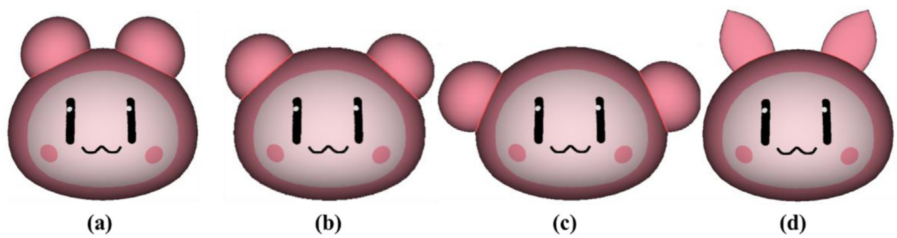


Fig. 16. Structure similarity of different models.

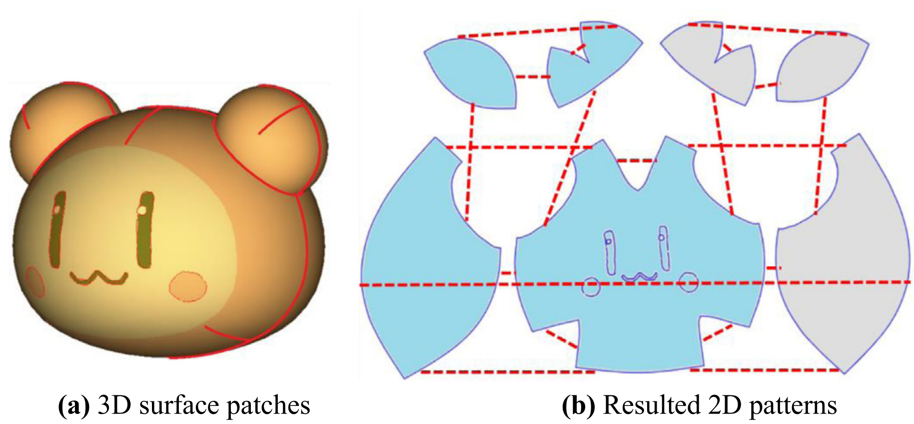


Fig. 17. An example of surface flattening.

target component Q, we check each template T with the same number of intersecting lines. The similarity of intersecting lines is calculated as follows:

$$stru\,(Q, T) = 100(1 - \frac{1}{2N_Q}\sum_{n=1}^{N_Q}\min(|C_Q^n C_T^m|)),\ n, m = 1, 2, \ldots, N_Q \tag{8}$$

where C_Q^n is the center of the intersecting line in Q, and C_T^m is the center of the intersecting line which is the closest to C_Q^n in T.

Structure similarity is used to judge whether two parts are structurally similar. Fig. 16 shows four head models of similar structures, the structural similarities between Fig. 16a and Fig. 16b, Fig. 16c, Fig. 16d are 85.87%, 68.16% and 96.94% respectively.

6. Pattern development and optimization

After retrieving the most similar template, seam lines will be automatically mapped on each component of a toy model. The generated seam lines divide the surface of the model into several surface patches, which can be flattened to 2D patterns of the toy model.

6.1. Surface flattening

As we know, a physical plush toy is sewn by several fabric pieces along the seam lines. To get 2D patterns from 3D surface patches, we adopt a flattening method [27], which is based on a mass–spring model to generate 2D patterns by unfolding 3D surfaces. A multilevel-mesh acceleration and boundary optimization are employed to make the flattening process fast and effective. After a 2D mesh is obtained, a polygon is generated based on the boundary of the 2D mesh, and a 2D pattern is obtained. Fig. 17a shows a toy model whose surface is divided into several surface patches by seam lines and Fig. 17b shows the flattened patterns. According to the sewing relations (connected by red dotted lines), a plush toy can be easily sewn by assembling fabric pieces.

6.2. Optimization of design schemes

The flattening results of 2D patterns are determined by the design of seam lines. If any seam lines mapped from the selected template are not reasonable, further modification is necessary to achieve a satisfactory design. One simple refinement method is to manually modify the seam lines. The user can easily adjust a control point of a seam line to the appropriate position to refine

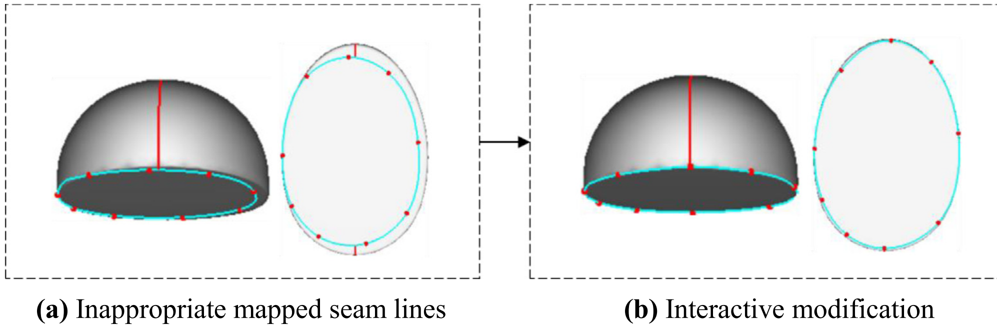


Fig. 18. Interactive modification of seam lines.

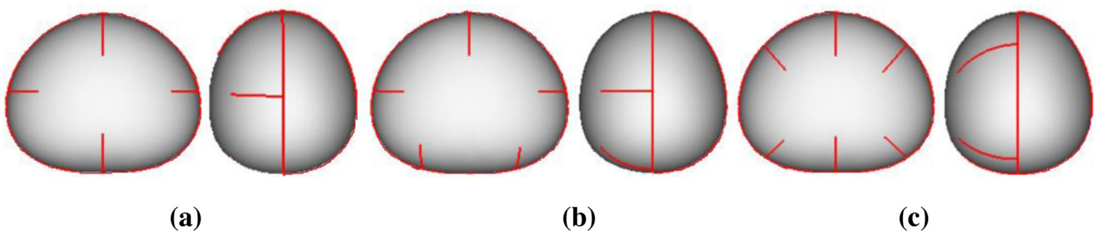


Fig. 19. Three design schemes of a head component.

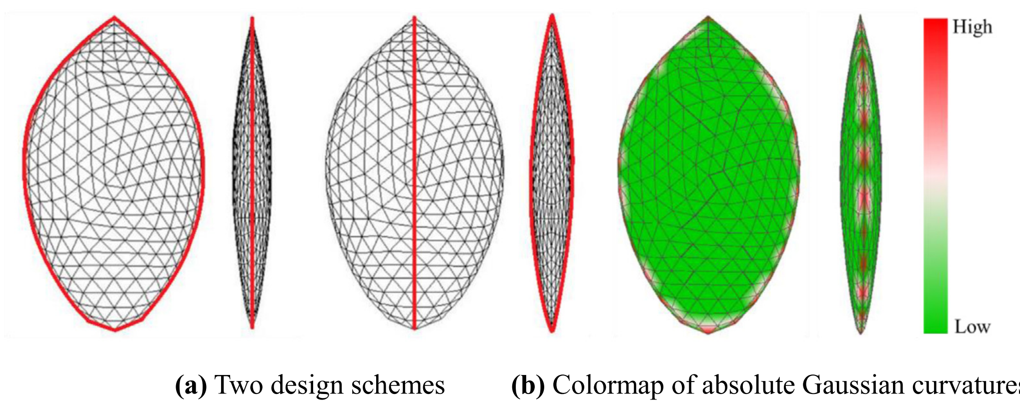


Fig. 20. Use Gaussian curvatures to select a reasonable design scheme. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

the line. In Fig. 18, the mapped seam line on the bottom of a foot component is inappropriate, since it is not along the sharp region. A reasonable seam line can be obtained by interactively moving its control points.

Usually, there exist several different design schemes that are available for one component. Fig. 19 shows a head component with different schemes. In practice, the designer chooses an appropriate scheme based on her/his own experience and the properties of the toy. In our method of 3D model retrieval, several most similar templates with different designs of seam lines can be found. It is not easy to choose the most suitable scheme.

To help the user to choose the best of several design schemes, we employ two methods. The first method is to calculate the stretch degree of flattened patterns. A mass–spring model [28] is used to calculate the deformation energy of a 2D mesh flattened from a 3D mesh. Normally a scheme with small deformation means that the design tends to be reasonable. For this method, the reasonability of a scheme can be judged only after all patterns are flattened.

An alternative method is to judge the reasonability of a scheme without flattening the patterns. Inspired by the paper [29] that tries to find the shortest path along points with high Gaussian curvatures to generate cutting lines, we use the mean absolute value of Gaussian curvature of seam lines to represent the stretch degree of each design scheme. Discrete Gaussian curvatures [30] are computed for the faces where the seam lines go through. A scheme with a relatively high absolute mean value $\overline{K}$ of Gaussian curvatures stands for a smaller stretch after flattening. Fig. 20 shows two different design schemes on the same component. The absolute mean value of Gaussian curvatures of the left scheme in Fig. 20a is 0.0068, and the right is 0.0029. Fig. 20b shows the colormap of absolute Gaussian curvatures of this component. It is obvious that the flattening stretch of the left scheme in Fig. 20a is smaller than the right scheme.

7. Experimental results

We have implemented our method using C++ and OpenGL and some experiments have been conducted. The platform of the computer we tested is on Windows 10 (64-bit) with GPU 1.60 GHz, 8 G memory.

In the experiments, we collected 13 kinds of animals and totally 90 models for testing, including 32 bears, 11 rabbits, 8 dogs, 3 tigers, 3 lions, 6 chickens, 5 horses, 6 cats, 1 cow, 2

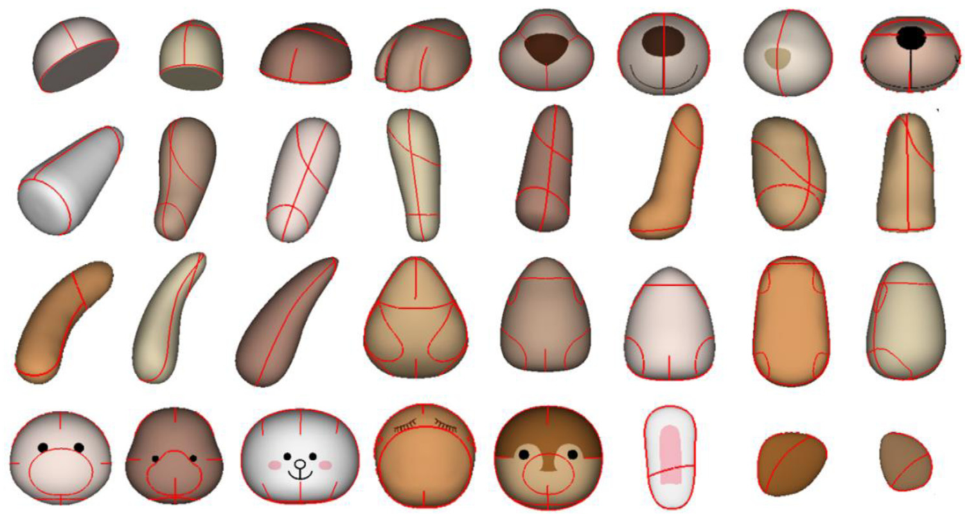


Fig. 21. Some components in the template library.

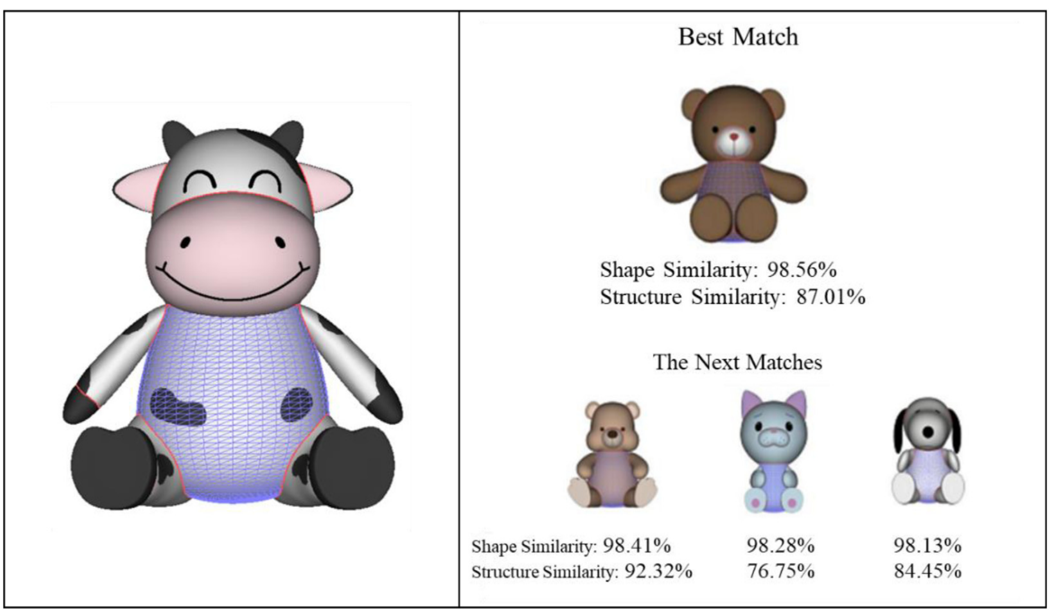


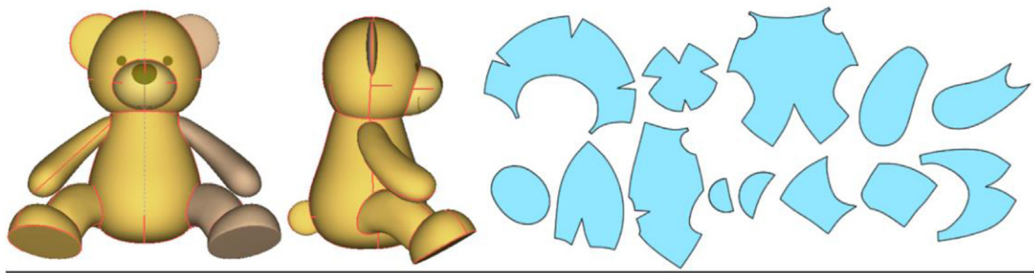
Fig. 22. Retrieved result of a component (Left: target component, right: retrieval result, all highlighted in blue). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

mice, 5 elephants, 7 pigs and 1 sheep. There are 656 component templates generated in the library, including 84 ears, 88 heads, 90 bodies, 90 hands, 90 legs, 64 palms, 72 tails and 78 noses. Fig. 21 shows some components in the library.

The operations of designing patterns for a 3D toy model are as follows. Firstly, input a 3D toy model into the program, the intersecting lines of all components are generated automatically. Next, choose one component of the model, and search the template library to find four most matched templates with similar shapes and structures (Fig. 22 shows the retrieval result of a target component, in which the shape similarities and structure similarities are displayed). Then select one template which is regarded as the most matched template and map its seam lines to the target component. The user can interactively modify the automatically generated seam lines if necessary. Repeat the above operations until all components are processed. At last, flattening all 3D surface patches to obtain patterns of the toy model.

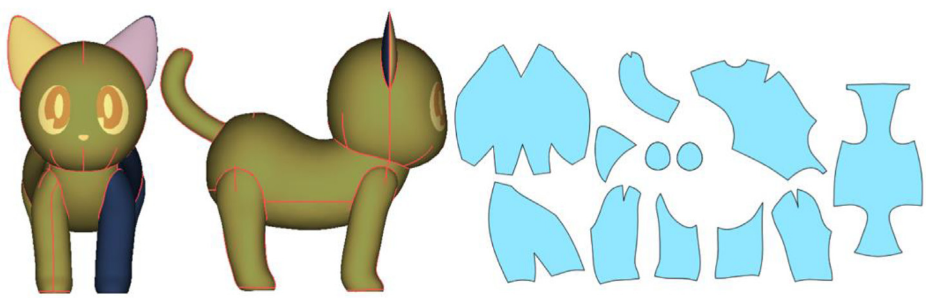
Experiments show the runtime of template retrieval for the component is less than 0.5s, and the runtime of seam lines mapping is less than 0.02s. To test the validity and feasibility of our method, we chose several toy models with different animal types and different postures, some of which have different anatomical structures. Fig. 23 shows the experimental results conducted by non-expert users after learning the operations of the program and basic knowledge of pattern design. In the table of each figure, the mapping template (highlighted in blue) corresponding to each selected component of the target model is displayed, and the degrees of shape similarity and structure similarity are shown. Our method can deal with symmetric and asymmetric toy models and Fig. 23f shows an asymmetric model.

We selected two models in Fig. 23a and b to manufacture physical toys. The final real toys were produced directly based on the resulted patterns without any modification. Fig. 24 shows the produced toys, and we can see that the real toys are very similar to the 3D models.



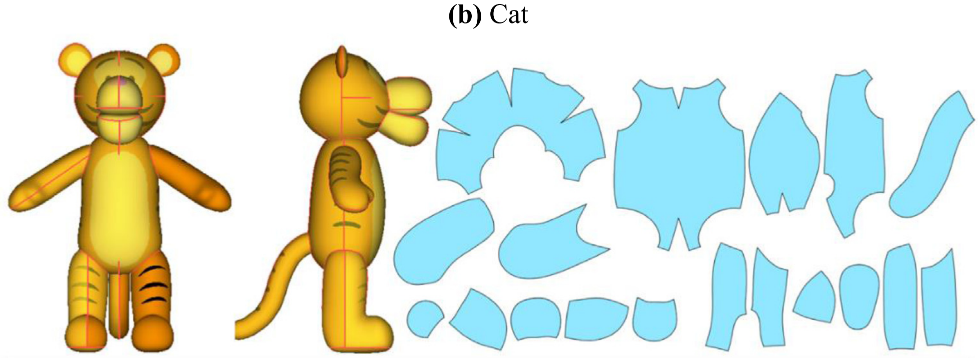
	Left ear	Head	Nose	Left hand	Body	Left leg	Left foot	Tail
Template mapped								
Shape Similarity	98.90%	99.10%	96.90%	97.19%	98.14%	93.95%	97.75%	98.14%
Structure Similarity	85.76%	93.69%	88.12%	91.52%	86.90%	84.35%	60.71%	84.83%

(a) Teddy bear



	Left ear	Head	Left front leg	Body	Left rear leg	Tail
Template mapped						
Shape Similarity	96.53%	96.12%	95.80%	93.66%	93.95%	88.23%
Structure Similarity	91.99%	78.92%	90.35%	89.68%	84.35%	92.99%

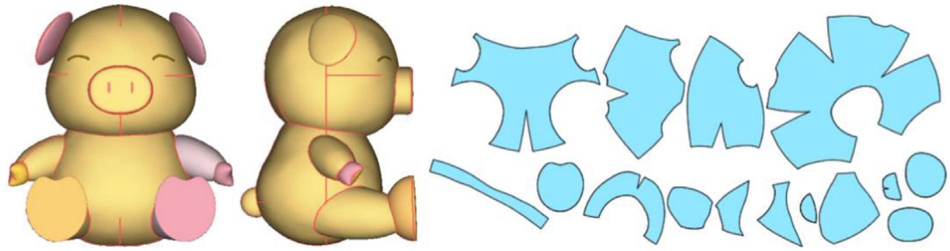
(b) Cat

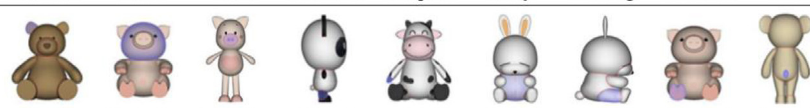


	Left ear	Head	Nose	Mouse	Left hand	Body	Left leg	Tail
Template mapped								
Shape Similarity	98.25%	95.56%	96.29%	88.43%	95.87%	96.41%	89.19%	89.65%
Structure Similarity	78.50%	81.51%	71.06%	62.78%	89.54%	96.54%	91.91%	97.76%

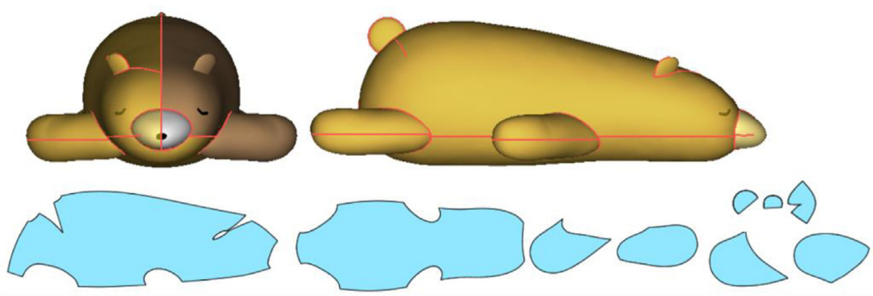
(c) Tiger

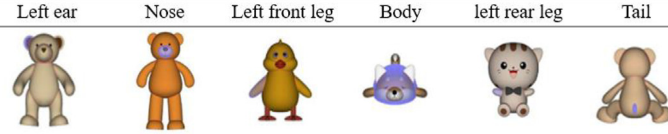
Fig. 23. Designed patterns of toys using component-based templates. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)



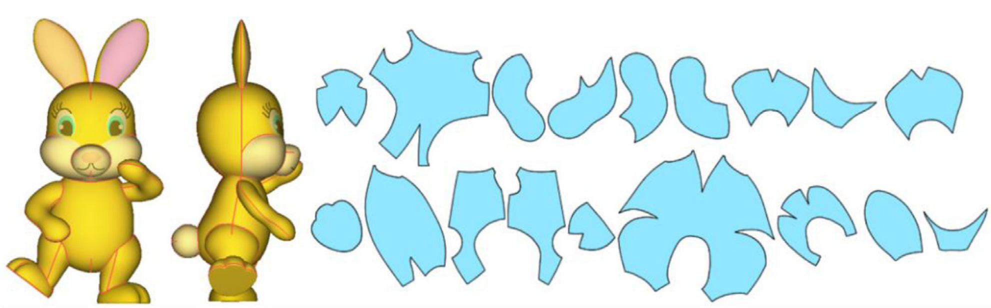
	Left ear	Head	Nose	Left hand	Left palm	Body	Left leg	Left foot	Tail
Template mapped									
Shape Similarity	97.98%	97.94%	97.69%	96.61%	92.15%	96.88%	97.51%	97.09%	99.11%
Structure Similarity	66.93%	78.40%	94.70%	82.26%	91.50%	87.72%	89.43%	93.58%	90.90%

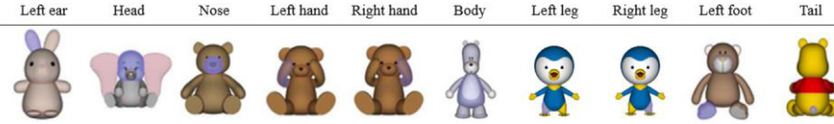
(d) Pig



	Left ear	Nose	Left front leg	Body	left rear leg	Tail
Template mapped						
Shape Similarity	98.00%	96.78%	95.82%	87.67%	92.87%	99.19%
Structure Similarity	85.40%	87.55%	77.32%	80.53%	81.55%	60.09%

(e) Sleeping bear



	Left ear	Head	Nose	Left hand	Right hand	Body	Left leg	Right leg	Left foot	Tail
Template mapped										
Shape Similarity	98.13%	96.02%	97.21%	90.10%	89.87%	97.13%	93.75%	92.49%	98.39%	99.94%
Structure Similarity	97.04%	78.10%	90.60%	89.59%	89.74%	92.12%	82.80%	80.95%	90.24%	78.99%

(f) Dancing rabbit

Fig. 23. (*continued*).

Fig. 24. Manufactured physical toys based on patterns designed by our method.

8. Conclusions

An efficient pattern design method for plush toys has been presented in this paper. The component-based template is introduced to reuse the knowledge of toy design and speed up the design process. With a template library, seam lines can be generated automatically on a 3D model according to the retrieved templates. The seam lines divide the surface of the 3D model into several surface patches, and the final 2D patterns are gotten by flattening the surface patches. The experimental results demonstrated that the proposed method is feasible and usable, and the produced results are satisfactory. The method makes the work of toy design simple, and the efficiency of design can be improved greatly. It is applicable to both professional designers and non-professionals.

The main contributions of the study are as follows:

(1) A component-based template is proposed to represent the knowledge of toy design, which makes it possible to reuse the knowledge and skills of professional designers and improve the efficiency of design.
(2) A component mapping method and 3D model retrieval are presented to build the correspondence of two components and map the seam lines of a template to the target component, so that the design knowledge is reused.

Some limitations exist in the current method. One limitation is that the method cannot deal with the toy model with all components merged into one mesh. Another limitation is that it is difficult to handle 3D models with very complex components at the current stage. If the data in the template library is not big enough, the automatically generated seam lines on a complicated component may not be satisfactory.

Future work remains to be done. The first is the automatic segmentation for a non-component-based 3D model. Our method can be applied to a non-component-based model if it is segmented and converted into a component-based model. The second is automatically identifying and classifying components. To obtain better results of pattern design for a toy, we should not only recognize the classification of the 3D toy models, but also the classification and characteristics of each component. The last is using artificial intelligence (AI) techniques to learn the knowledge of pattern design to deal with complex models. Nowadays, the artificial neural network has performed well in the processing of unstructured data such as natural language and computer vision. For the knowledge of pattern design, it is also possible to use AI to learn, analyze, and make decisions to assist users to realize the intelligent design.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work is supported in part by the National Natural Science Foundation of China under Grant 61972340, 61732015 and 61972220, and in part by the Zhejiang Province through the Key Research and Development Program, China under Grant 2018C01090.

References

[1] Igarashi T, Matsuoka S, Tanaka H. Teddy: A sketching interface for 3d freeform design. In: ACM SIGGRAPH. 1999.
[2] Liu Y-J, Ma C-X, Zhang D-L. EasyToy: plush toy design using editable sketching curves. IEEE Comput Graph Appl 2009;31(2):49–57. http://dx.doi.org/10.1109/MCG.2009.147.
[3] Funkhouser T, Kazhdan M, Shilane P, Min P, Kiefer W, Tal A, et al. Modeling by example. In: ACM transactions on graphics. ACM; 2004, http://dx.doi.org/10.1145/1015706.1015775.
[4] Chaudhuri S, Kalogerakis E, Guibas L, Koltun V. Probabilistic reasoning for assembly-based 3D modeling. In: ACM transactions on graphics. ACM; 2011, http://dx.doi.org/10.1145/1964921.1964930.
[5] Jain A, Thormählen T, Ritschel T, Seidel HP. Exploring shape variations by 3d-model decomposition and part-based recombination. In: Computer graphics forum. Wiley Online Library; 2012, http://dx.doi.org/10.1111/j.1467-8659.2012.03042.x.
[6] Chaudhuri S, Koltun V. Data-driven suggestions for creativity support in 3D modeling. In: ACM transactions on graphics. ACM; 2010, http://dx.doi.org/10.1145/1882261.1866205.
[7] Kim VG, Li W, Mitra NJ, Chaudhuri S, DiVerdi S, Funkhouser T. Learning part-based templates from large collections of 3D shapes. ACM Trans Graph 2013;32(4):70. http://dx.doi.org/10.1145/2461912.2461933.
[8] Xue T, Liu J, Tang X. Example-based 3D object reconstruction from line drawings. In: 2012 IEEE conference on computer vision and pattern recognition. IEEE; 2012, http://dx.doi.org/10.1109/cvpr.2012.6247689.
[9] Kwok T-H, Zhang Y-Q, Wang CC, Liu Y-J, Tang K. Styling evolution for tight-fitting garments. IEEE Trans Vis Comput Graph 2015;22(5):1580–91. http://dx.doi.org/10.1109/tvcg.2015.2446472.
[10] Mori Y, Igarashi T. Pillow: interactive pattern design for plush toys. DVD Publication at SIGGRAPH; 2006, http://dx.doi.org/10.1145/1179849.1179942.
[11] Mori Y, Igarashi T. Plushie: an interactive design system for plush toys. ACM Trans Graph 2007;26(3):45. http://dx.doi.org/10.1145/1276377.1276433.
[12] Skouras M, Thomaszewski B, Kaufmann P, Garg A, Bickel B, Grinspun E, et al. Designing inflatable structures. ACM Trans Graph 2014;33(4):1–10. http://dx.doi.org/10.1145/2601097.2601166.

[13] Wang J, Lu G, Li W, Chen L, Sakaguti Y. Interactive 3D garment design with constrained contour curves and style curves. Comput Aided Des 2009;41(9):614–25. http://dx.doi.org/10.1016/j.cad.2009.04.009.

[14] Wang CC, Tang K. Pattern computation for compression garment by a physical/geometric approach. Comput Aided Des 2010;42(2):78–86. http://dx.doi.org/10.1016/j.cad.2009.02.018.

[15] Umetani N, Kaufman DM, Igarashi T, Grinspun E. Sensitive couture for interactive garment modeling and editing. ACM Trans Graph 2011;30(4):90. http://dx.doi.org/10.1145/2010324.1964985.

[16] Paquet E, Rioux M, Murching A, Naveen T, Tabatabai A. Description of shape information for 2-D and 3-D objects. Signal Process Image Commun 2000;16(1–2):103–22. http://dx.doi.org/10.1016/s0923-5965(00)00020-5.

[17] Ohbuchi R, Takei T. Shape similarity comparison of 3D models using alpha shapes. In: 11th pacific conference on computer graphics and applications, 2003. proceedings. IEEE; 2003, http://dx.doi.org/10.1109/pccga.2003.1238271.

[18] Vranic DV, Saupe D. 3D model retrieval (Ph.D. thesis), University of Leipzig; 2004.

[19] Liu A-A, Nie W-Z, Gao Y, Su Y-T. Multi-modal clique-graph matching for view-based 3D model retrieval. IEEE Trans Image Process 2016;25(5):2103–16. http://dx.doi.org/10.1109/tip.2016.2540802.

[20] Wang F, Kang L, Li Y. Sketch-based 3d shape retrieval using convolutional neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2015, http://dx.doi.org/10.1109/cvpr.2015.7298797.

[21] Nie W-Z, Liu A-A, Su Y-T. Representation I, 3D object retrieval based on sparse coding in weak supervision. J Vis Commun 2016;37:40–5. http://dx.doi.org/10.1016/j.jvcir.2015.06.011.

[22] Theologou P, Pratikakis I, Theoharis T. Part-based 3D object retrieval via multi-label optimization. Comput Vis 2017;160:148–57. http://dx.doi.org/10.1016/j.cviu.2017.04.006.

[23] Wang Z, Lu F. VoxSegNet: Volumetric CNNs for semantic part segmentation of 3D shapes. IEEE Trans Vis Comput Graph 2019.

[24] Mo K, Zhu S, Chang AX, Yi L, Tripathi S, Guibas LJ, et al. Partnet: A large-scale benchmark for fine-grained and hierarchical part-level 3d object understanding. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2019.

[25] McKinley S, Levine M. Cubic spline interpolation. College Redwoods 1998;45(1):1049–60. http://dx.doi.org/10.1142/9789812813381_0003.

[26] Alexa M. Merging polyhedral shapes with scattered features. In: Proceedings shape modeling international'99. international conference on shape modeling and applications. IEEE; 1999, http://dx.doi.org/10.1007/pl00007211.

[27] Zhang D, Li J, Wang J. Design patterns of soft products using surface flattening. J Comput Inf Sci Eng 2018;18(2):021011. http://dx.doi.org/10.1115/1.4039476.

[28] Wang CC, Smith SS, Yuen MM. Surface flattening based on energy model. Comput Aided Des 2002;34(11):823–33. http://dx.doi.org/10.1016/s0010-4485(01)00150-6.

[29] Wang CC, Wang Y, Tang K, Yuen MM. Reduce the stretch in surface flattening by finding cutting paths to the surface boundary. Comput Aided Des 2004;36(8):665–77. http://dx.doi.org/10.1016/s0010-4485(03)00024-1.

[30] Botsch M, Pauly M, Kobbelt L, Alliez P, Lévy B, Bischoff S, et al. Geometric modeling based on polygonal meshes. 2007, http://dx.doi.org/10.1145/1281500.1281640.