Journal of Zhejiang University-SCIENCE C (Computers & Electronics)
ISSN 1869-1951 (Print); ISSN 1869-196X (Online)
www.zju.edu.cn/jzus; www.springerlink.com
E-mail: jzus@zju.edu.cn

A controllable stitch layout strategy for random needle embroidery*

Jie ZHOU†, Zheng-xing SUN†‡, Ke-wei YANG

(*State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210046, China*)

†E-mail: makey537@163.com; szx@nju.edu.cn

Received Mar. 18, 2014; Revision accepted Aug. 2, 2014; Crosschecked Aug. 19, 2014

Abstract: Random needle embroidery (RNE) is a graceful art enrolled in the world intangible cultural heritage. In this paper, we study the stitch layout problem and propose a controllable stitch layout strategy for RNE. Using our method, a user can easily change the layout styles by adjusting several high-level layout parameters. This approach has three main features: firstly, a stitch layout rule containing low-level stitch attributes and high-level layout parameters is designed; secondly, a stitch neighborhood graph is built for each region to model the spatial relationship among stitches; thirdly, different stitch attributes (orientations, lengths, and colors) are controlled using different reaction-diffusion processes based on a stitch neighborhood graph. Moreover, our method supports the user in changing the stitch orientation layout by drawing guide curves interactively. The experimental results show its capability for reflecting various stitch layout styles and flexibility for user interaction.

Key words: Random needle embroidery (RNE), Stitch style, Stitch layout, Stitch neighborhood graph, Reaction diffusion
doi:10.1631/jzus.C1400099 **Document code:** A **CLC number:** TP391

1 Introduction

Random needle embroidery (RNE), which is a graceful embroidery art originating in Changzhou, Jiangsu, China, has been enrolled in the world intangible cultural heritage. It creates an artwork by stitching thousands of intersecting threads with different lengths, different orientations, and vivid colors into a base cloth. Photographs of various subjects (e.g., portraits, animals, scenery, and oil paintings) can be used as embroidery drafts. Generally, RNE artworks are more impressive and expensive than traditional embroidery artworks for their various lively stitching styles, rich thread colors, stereoscopic feeling, and contrast among neighboring threads thanks to the creative inspirations from the combination of western oil painting and Chinese embroidery theories. This combination precisely increases the difficulty in training qualified embroiderers and the time cost in completing the final artwork. It is valuable to construct a reasonable technical framework to enhance common users' visual experiences and assist embroiderers in reality. The former work is actually what image-based artistic rendering methods (Zeng *et al.*, 2009; Xu *et al.*, 2010) are concerned about, namely, digitizing various art styles using the graphics technology. Moreover, our further goal is to increase the production efficiency in final machine embroidery. Existing embroidery software tools are not suitable for RNE due to the difference in stitching styles. In fact, the research on RNE involves several key issues: how to extract the image content, how to model various stitching styles, how to simulate its artistic effect, and how to generate the stitch sequence used in machine embroidery. Obviously, how to model various stitching styles is the basic and core issue.

‡ Corresponding author

* Project supported by the National Natural Science Foundation of China (Nos. 61272219, 61100110, and 61321491), the National High-Tech R&D Program (863) of China (No. 2007AA01Z334), the Key Projects Innovation Fund of State Key Laboratory (No. ZZKT2013A12), the Program for New Century Excellent Talents in Universities, China (No. NCET04-04605), the Science and Technology Program of Jiangsu Province (Nos. BE2010072, BE2011058, and BY2012190), and the Scientific Research Foundation of Graduate School of Nanjing University (No. 2012CL21), China

To be specific, this work attempts to convert a source image into a combination of intersecting stitches with different positions, orientations, lengths, and colors, where we call this combination problem a stitch layout. There is little in the literature about traditional embroidery or RNE. Chen *et al*. (2012) presented several stitch placement methods for traditional embroidery. However, this method uses simple vector images rather than arbitrary raster images as input. Furthermore, intersecting stitches are forbidden among the traditional embroidery stitch layout rules (e.g., long short stitches and satin stitch), and these rules place stitches relying more on shape information (e.g., edge and contour). Different from this, the textural and stereoscopic feelings conveyed by RNE stitch layout rules are more attractive due to the flexible use of visual information (color and texture). In addition, the stitches must be intersected with the neighboring stitches to depict different objects, no matter what layout forms. Our previous work (Chen *et al*., 2011) firstly proposed the research on RNE and confirmed its feasibility. A simple trial-and-error algorithm was adopted to fill stitches until a predefined coverage rate was reached. It is essentially a sliding-window brushing operation from top-left to bottom-right. Chen *et al*. (2011) regarded the whole image as a region and thus did not consider various combination forms of different stitch attributes (e.g., orientation, length, and color), which influence the layout result greatly. Moreover, this method cannot allow the user to control the global layout and local contrasts of stitches conveniently. Based on the above analysis, challenges are posed for an RNE stitch layout in the following ways:

First, we should respect the inherent characteristics of RNE. For example, the stitches must be intersected with the neighboring stitches to depict different objects, and different layout rules for stitches convey different feelings about objects. However, there is no manual or handbook definitely describing these rules, unlike traditional embroidery manuals (Chen *et al*., 2012). Therefore, the techniques modeling the stitch layout problem for RNE need to be studied. This model should be concerned with the correlation of the stitches, and how to intersect and layout them in a vector space.

Second, the strategy that controls the global stitch layout appearance should agree with some usual conventions associated with RNE. In fact, artists usually adopt horizontal or vertical intersecting form stitches for background regions with a single tone like the sky, along textural or growth directions, intersecting form stitches for regions like plumage, petals, or hair, completely random intersecting form stitches for some flat objects with low gradient magnitude values, and so on. Furthermore, it is a user-originality driven course to create artworks for RNE artists. A convenient control of the global stitch layout supporting user interaction is appreciated. Adjusting parameters (Xu *et al.*, 2010; Huang *et al.*, 2011) and drawing sketch (Xu *et al.*, 2009a; 2009b; 2013; O'Donovan and Hertzmann, 2012) are both widely-used and naturally interactive modes.

We study the above problems and propose a controllable stitch layout strategy for RNE. The main features of this approach include: (1) A stitch layout model design which involves low-level stitch attributes, low-level intersecting-stitch attributes, and high-level stitch layout parameters, to express different stitch layout styles. The high-level parameters are composed of density and five diffusion factors for different stitch attributes (orientation, length, color hue, color saturation, and color lightness). (2) A stitch neighborhood graph to model the relationship among neighboring stitches. (3) A stitch layout process based on a stitch neighborhood graph via five diffusion factors and a reaction-diffusion algorithm, enabling global layout control utilizing image information (orientation field, salience map, and image grids partition) or user interaction, and simultaneously the local contrast control.

2 Related works

In the computer graphics literature, a significant number of works aimed at simulating the appearance of various kinds of textiles, including batik (Wyvill *et al.*, 2004), knitwear (Kaldor *et al.*, 2008), and woven materials (Adabala *et al.*, 2003). However, all these works focus on the physical medium simulation, not on the layout problem.

The problem of this work, as a new art style, can be related to a newly developing field—artistic rendering, which includes various algorithms for different art styles, such as stippling (Deussen *et al.*, 2000; Secord, 2002; Martín *et al.*, 2010), mosaic (Hausner, 2001), pen-and-ink illustration (Salisbury *et al.*, 1994), maze (Xu and Kaplan, 2007), watercolor (Curtis *et al.*, 1997), ASCII art (Xu *et al.*, 2010), Op art (Inglis *et al.*, 2012), and oil-painting (Bratkova *et al.*, 2009; Zeng *et al.*, 2009; Huang *et al.*, 2011). These works can be classified as filter- and stroke-based methods. Filter-based methods directly make the image, with the specified visual effect, by image processing operators on pixels, while stroke-based methods build a stroke-layer, which is composed of numerous strokes and each stroke has its individual parameters or attributes, between the source and target images. Regarding one stitch as a single stroke, we can refer to the stroke-based methods, since all the stitches embroidered in cloth need be calculated for the final machine embroidery process and their layout rules should be researched.

The layout problem is an important issue among the stroke-based methods. It concerns how to arrange or place specific elements or primitives reasonably, such as parallel lines (Inglis *et al.*, 2012), rectangulars (Hausner, 2001), brush (Zeng *et al.*, 2009), deformable templates (Peng *et al.*, 2014), 3D model (Yang *et al.*, 2013), or graph visualization (Yuan *et al.*, 2012; Dogrusoz *et al.*, 2013; Gansner *et al.*, 2013).

There are mainly two kinds of layout strategies: greedy and optimization-based (Hertzmann, 2003). In greedy methods, at each step the algorithm determines the current stroke according to certain goals or image features (Litwinowicz, 1997; Hays and Essa, 2004; Lu *et al.*, 2010). The optimization-based methods compute the entire strokes together to achieve a global optimal energy or desired statistics. Theoretically, the optimization-based methods (Gansner *et al.*, 2013; Yang *et al.*, 2013) have the potential to outperform the greedy methods, since they can explicitly model the interaction among elements and supply the high-order control. This approach belongs to the optimization-based class.

Stroke layout determines the attributes of a stroke automatically or by user interaction. Generally, these attributes are (1) position and density, (2) orientation, (3) size, and (4) color and texture.

For position and density, Kalnins *et al.* (2002) required the user to place a few strokes. These strokes are used as templates and then replicated at other locations by calculating the distance of the template strokes from feature lines. Others took a more automated approach. Litwinowicz (1997) placed strokes at alternate pixel locations. Park and Yoon (2008) placed strokes at locations where a randomly generated value exceeds the probability threshold. Shiraishi and Yamaguchi (2000) attempted to estimate the area of the strokes by generating an intensity image using image moments. Mao *et al.* (2002) determined the density of strokes by generating noise at a density based on the tone of the underlying image. Dithering methods (Gamito and Maddock, 2009) are also used to generate the locations of strokes. These existing methods directly determine positions in a canvas space. In RNE, stitches positions should be determined in a cloth space and stitches need to be distributed evenly. Xu *et al.* (2010) only placed ASCII characters along region boundaries. However, RNE uses a large number of stitches to fill regions.

For orientation, Litwinowicz (1997), Collomosse and Hall (2002), and Santella and DeCarlo (2002) used gradients to align their strokes. Hays and Essa (2004) and Huang *et al.* (2011) used a radial basis function (RBF) to interpolate orientation by the strongest gradients. These methods work well in high-frequency regions where there is much variation in color. In regions where the colors of neighboring pixels are similar, the gradient vector tends to point in an arbitrary direction. Shiraishi and Yamaguchi (2000) used image moments to determine an angle for stroke orientation. Zeng *et al.* (2009) computed orientations using a Markov random field framework. Ding *et al.* (2012) placed streamlines using 2D vector fields. All the above methods just use a fixed image gradient or orientation field to determine the stroke orientation, and thus cannot supply an intuitional control of the stitch orientation layout. Inglis *et al.* (2012) allowed only a fixed number of orientations to place parallel lines and curves for Op art; hence, it is also not suitable for our problem.

For size, Litwinowicz (1997) generated short strokes to give the output an impressionistic style. Collomosse and Hall (2002) determined the shape of a stroke using a super-quadric equation. Mao *et al.* (2001) and Yamamoto *et al.* (2004) controlled the length of strokes by varying the length of the convolution kernel used in line integral convolution (LIC). These methods do not use straight lines as a basic element and thus cannot be used in our problem.

Inglis *et al.* (2012) just used parallel lines to fill a rectangle region, where parallel lines are cut off or connected along the region boundary. However, we need to express different objects by intersecting stitches with different lengths that are usually much smaller than the region size.

For color and texture, many methods just sample the reference image's color or use average color as stroke color (Luft and Deussen, 2006; Bratkova *et al.*, 2009). Users cannot control the stroke colors conveniently. Some other art styles (Deussen *et al.*, 2000; Secord, 2002; Xu *et al.*, 2010; Inglis *et al.*, 2012) even use only black and white. Xie *et al.* (2010) generated one brushstroke color procedurally. The footprints of the brushstroke are divided into six segments. Each segment is assigned a texture and color from a sample of six styles. However, it processes only one stroke in the Sumi-e painting. This approach uses three reaction-diffusion processes to create vivid, vibrant, or high-contrast colors for emphasizing objects. It is easy for users to feel these variations or contrasts of hue, saturation, and lightness.

A reaction-diffusion algorithm was originally used to model the physical processes of the chemical reaction between substances and their diffusion in space. It has been used in biological patterns (Kondo and Miura, 2010), 3D model texture generation (Turk, 1991; Sanderson *et al.*, 2006), or image processing (Perona, 1998; Chen and Zhu, 2006). This diffusion mechanism can control the global and local schemas easily, utilizing neighboring information. In this paper, we build a topological structure to model the relationship of all stitches and use reaction-diffusion to control their layout based on this structure.

3 Stitch layout model design

In RNE, artists usually depict the appearance of different objects by intentionally embroidering one stitch to intersect with other stitches, which is quite different from the traditional embroidery where non-overlapping or end-to-end stitches are commonly adopted. Different intersecting forms show different stitch styles. To express these so-called stitch styles, we design a parameterized stitch layout model that includes low-level stitch attributes, low-level intersecting-stitch attributes, and high-level stitch layout parameters (Fig. 1).

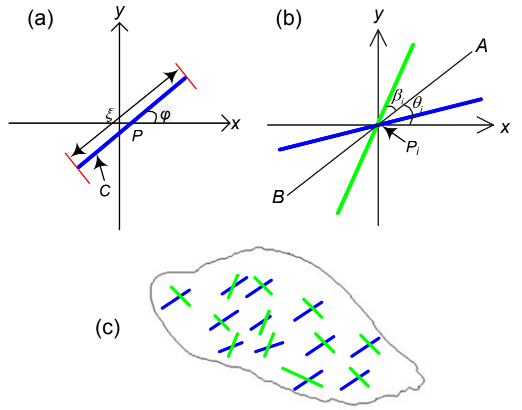

Fig. 1 Stitch layout model for the random needle embroidery (RNE)
(a) A stitch with center coordinate P, length ξ, orientation φ, and color C; (b) An intersecting-stitch with center coordinate P_i, intersection orientation θ_i, and intersection angle β_i; (c) Global layout of intersecting stitches. Each intersecting-stitch (b) includes two stitches (a). The global layout (c) is a combination of various intersecting-stitches

Stitch: One stitch is the basic element of RNE to compose its artworks. Appropriately, we first define the stitch attributes. As Fig. 1a shows, a stitch contains the following attributes: center coordinate P, length ξ, orientation φ, and color C.

Intersecting-stitch: In RNE, there must be an intersection among neighboring stitches. There are many kinds of intersecting forms, such as horizontal intersecting, vertical intersecting, smooth intersecting, or chaotic intersecting. The orientations of the stitches created by smooth intersecting usually follow some fixed directions or beneath the texture information, such as water flow or leaf growth; however, the chaotic cross is quite the opposite. That is, all the stitches are in a mess and have no unified direction. Among all these cross forms, the stitch orientation is a very important factor for depicting different styles in RNE. Based on this characteristic, we further define the intersecting form of two stitches. Fig. 1b shows intersecting-stitch attributes: center coordinate P_i, intersection orientation θ_i, and intersection angle β_i.

Layout: As mentioned before, the stitch layout is one basic problem deserving research in RNE. However, there are no specific instructions about RNE layout rules compared with the traditional embroidery. Hence, we design a layout model which is a collection of intersecting-stitches with different positions, orientations, lengths, and colors (Fig. 1c). Considering the truth that no single stitch but the stitches combined together express tone, texture, or other object characteristics, we build a stitch neighborhood graph for all stitches of each region to model their spatial relationship (Section 4.3). Based on this, we design our high-level layout control parameters: density ρ, orientation diffusion factor λ_θ, length diffusion factor λ_ξ, hue diffusion factor λ_h, saturation diffusion factor λ_s, and lightness diffusion factor λ_v. The overall coverage is controlled by ρ (Section 4.2); λ_θ and λ_ξ control global stitches' orientations and lengths; λ_h, λ_s, and λ_v control their colors (Section 4.4). Different reaction-diffusion processes are adopted to control different stitches attributes via diffusion factors. Notice that a user can adjust all these high-level parameters for each region.

4 Stitch layout control

Essentially, RNE artwork is a collection of many stitches, each of which contains two endpoints' coordinates and its color. For each stitch, its coordinates should agree with the coordinates in embroidery cloth (the unit is mm). However, the image coordinates are quite different (the unit is pixel). Therefore, our method first generates the stitch collection in the cloth space, and then executes the rendering process of the stitches in the image space.

Generally, the stitch styles of different objects (e.g., grass, sky, petal, hair, and skin) are different in RNE, and correspondingly in the stitch layout results. Firstly, we segment the input image into different regions and vectorize their boundaries. Then, different high-level layout parameters are applied to different regions, and correspondingly different stitch collections are generated. For each region, the final layout of stitches filling it is computed as follows:

1. Stitch layout initialization: place intersecting-stitches uniformly in embroidery cloth according to the 'density' parameter.

2. Stitch neighborhood graph: connect neighboring intersecting-stitches to construct a topological structure.

3. Stitch attributes processes: process intersecting-stitch attributes on the graph, for stitch orientations, lengths, and colors, respectively.

4.1 Image segmentation and region vectorization

We adopt the image-parsing method proposed by Zeng *et al.* (2009) to segment the source image into several regions. To place stitches for each region in embroidery cloth, the boundary points of each region are parameterized by a cubic B-spline curve. Fig. 2a shows a binary mask of the input region (white) and Fig. 2b shows the original boundary and the vectorized boundary parameterized by cubic B-spline curve control points.

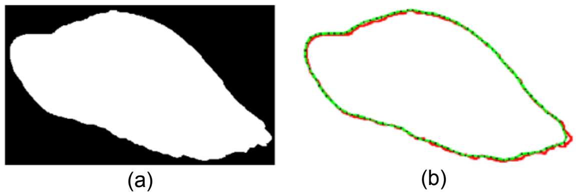

Fig. 2 Region boundary vectorization

(a) Region mask (white); (b) Original (red curve) and vectorized (green curve with dots) boundaries. References to color refer to the online version of this figure

4.2 Stitch layout initialization

For each region extracted in Section 4.1, we place a number of intersecting-stitches in the corresponding region of embroidery cloth, and each stitch's attributes are set random values, since the embroidery cloth should be covered uniformly, namely, all these stitches should be well-distributed, avoiding the phenomenon of locally too dense or too sparse. To ensure the overall uniform coverage, we adopt the point distributing methods, which are widely used in 2D and 3D rendering, to determine the initial position of each intersecting-stitch. In the 2D case, Poisson-disk sampling (Vanderhaeghe *et al.*, 2007; Gamito and Maddock, 2009) is an easy but time-consuming method. To some extent, stippling (Secord, 2002; Martín *et al.*, 2010) is essentially a kind of point distributing method. Meanwhile, these methods are able to control the denseness of all the points. However, many of them are based on the image space or use the pixel as the point position. These sampling algorithms are virtually a discrete process, hence not suitable for our process. As mentioned before, stitches are distributed in the embroidery cloth space, whose coordinates are continuous. In addition, a more reasonable way to control the density of the stitch layout is desired. We adopt the recursive Wang tile method (Kopf *et al.*, 2006) to generate the positions of initial intersecting-stitches. Notice that this method can quickly compute any number of blue-noise points whose coordinates are real float value, not pixel value. It can also control density by a density map.

According to the 'density' parameter (Section 3) of high-level layout, we sample the given number of intersecting-stitch positions in the region evenly via the recursive Wang tile method (Kopf *et al.*, 2006). Fig. 3 shows that different 'density' values lead to different coverage rates. Meanwhile, the overall distribution is maintained as uniform.

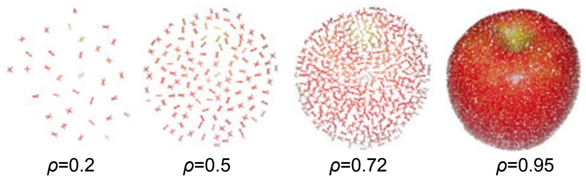

Fig. 3 Stitch layout with different densities, ρ, showing different coverage rates

4.3 Stitch neighborhood graph

Stitches in RNE work collectively, similar to pen-and-ink (Salisbury *et al.*, 1994). Namely, no single stitch is of critical importance in general, although every stitch contributes both tone and texture; instead, a lot of stitches work together to express abundant information, such as overall tone or textural feeling. Therefore, it is necessary to build a topological relationship among all the stitches.

To control the layout of all stitches effectively and process their attributes to match a desired statistic, we construct a Markov stitch neighborhood graph, where nodes are intersecting-stitch positions and edges connect each node with its neighboring nodes. Treating each intersecting-stitch's position as a node of the graph, we build the neighborhood graph based on these positions and the orientation field. Assuming that the source image size (in pixel) is $W_{\text{pixel}}\times H_{\text{pixel}}$ and the embroidery cloth size (in mm) is $W_{\text{mm}}\times H_{\text{mm}}$, our approach is as follows:

1. Compute an orientation field $\boldsymbol{\Theta}$ using the edge tangent flow method (Kang *et al.*, 2007). Notice that $\boldsymbol{\Theta}$ is a 2D matrix with a size the same as the size of the source image.

2. Extract each node's position (x, y), calculating $(x', y')=(\lfloor W_{\text{pixel}}x/W_{\text{mm}}\rfloor, \lfloor H_{\text{pixel}}y/H_{\text{mm}}\rfloor)$ as the corresponding image position; initialize each node's orientation to the reference orientation θ^* in $\boldsymbol{\Theta}$.

3. Construct local 2D Cartesian coordinates whose origin is anchored at each node's position; x-axis is aligned with the node's orientation θ^*, and y-axis is orthogonal to it.

4. Connect the four edges from the node to its nearest neighbor in each of the four quadrants. Notice that the nodes that are near region boundaries or image edges may not have neighbors in some quadrant.

Fig. 4 shows an example containing two regions. The 'density' parameter is 0.72 for both regions. Each graph node denotes each intersecting-stitch and each graph edge denotes the neighboring relation of two intersecting-stitches.

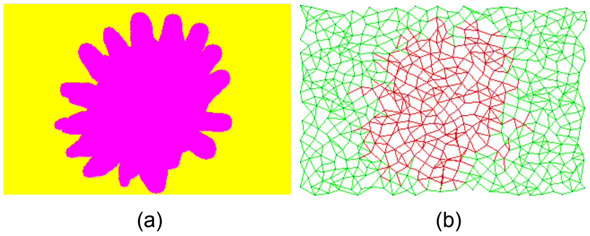

Fig. 4 Stitch neighborhood graph

Two graphs (b) are built for two regions (a) separately

4.4 Stitch attributes processes

In this stage, our approach controls all the stitches attributes of the current region, including orientations, lengths, and colors. All these attributes affect the global layout styles. Based on the stitch neighborhood graph, we adopt stochastic reaction-diffusion equations, which process these attributes among neighboring stitches to reduce or enhance their contrasts, and the reaction preserves information from the source image.

4.4.1 Intersecting-stitch orientations

The reaction-diffusion equation

$$\frac{\mathrm{d}\theta}{\mathrm{d}t}=R(\theta)+\lambda_\theta D(\theta)+\varepsilon_\theta \tag{1}$$

is adopted to propagate information across the stitch neighborhood graph to compute orientations of all the stitches, where $R(\theta)$ is the local reaction term, $D(\theta)$ is the local diffusion term, λ_θ is the diffusion factor, and ε_θ is the local random term. ε_θ is a small stochastic value at each iteration to express more natural randomness in reality. The user can adjust λ_θ to obtain different layout styles.

The local reaction term applies the persistent external force to the whole reaction-diffusion iterations. Namely, it plays the role of controlling the so-called global rule that the overall layout of all the stitches should obey. We use

$$R(\theta)=\sin(\theta^*-\theta) \quad (2)$$

to control the overall layout of intersecting-stitches, where θ^* is the corresponding external orientation force that can be acquired from the reference orientation field $\boldsymbol{\Theta}$ (Section 4.3).

Inspired by O'Donovan and Hertzmann (2012), our approach can take user interactions as an external orientation force (Fig. 5). The orientations of any initial stitches can be diffused to follow several user guide curves. Notice that these curves are also parameterized by cubic B-spline curve control points. We select the stitches whose minimum spatial distance to all the user guide curves is under the specified threshold and set their orientations to be the tangential direction of the nearest control point on the guide curve.

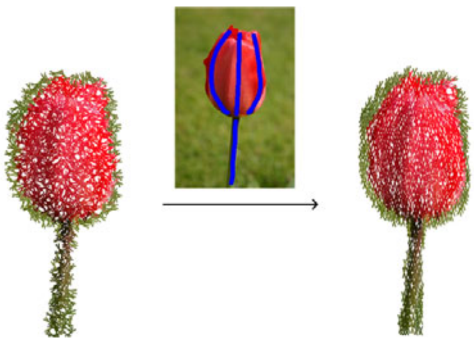

Fig. 5 User interactive guide curves

Different from the local reaction term, the local diffusion term applies the force to the local neighbors, and thus propagates the orientation value across the stitch neighborhood graph. It plays the role of controlling the local layout of each intersecting-stitch according to its neighbors in the graph. We adopt the orientation diffusion (Perona, 1998) term since θ is periodic over intervals of 2π,

$$D(\theta)=\sum_n \omega_n \sin(\theta_n-\theta), \quad (3)$$

where θ_n is each neighbor's orientation of the currently processing intersecting-stitch and ω_n is the weight inversely proportional to their spatial distance.

The diffusion factor λ_θ controls the diffusion degree of the local diffusion term. Specifically, when $\lambda_\theta>0$, the local diffusion term rotates each intersecting-stitch's orientation toward those of its neighbors to make them similar, and thus shows a smooth layout style with approximately similar orientations. When $\lambda_\theta<0$, it forces diffusion to rotate orientations of neighboring intersecting-stitches away from each other, and thus leads to a chaotic layout style with quite different orientations.

The process in Eq. (1), if without ε_θ, essentially optimizes the Markov field energy

$$E(\theta)=\sum_i \varphi(\theta_i-\theta_i^*)+\lambda_\theta\sum_i\sum_{j\in N(i)}\omega_{ij}\varphi(\theta_i-\theta_j), \quad (4)$$

where $\varphi(\cdot)=1-\cos(\cdot)$ is the kernel function, and each weight ω_{ij} is inversely proportional to the spatial distance between neighboring intersecting-stitches i and j.

In the process of iterations, the energy value trends towards convergence, and thus the orientations of all intersecting-stitches tend to a stable status where there is fairly small variation. After that, we update all these orientations. The stitch neighborhood graph construction and reaction-diffusion process are both very fast. Although the speed is obviously affected by the number of intersecting-stitches, it is still a linear increase. In our experiments, 200 iterations work well enough. In addition, both procedures can be parallelized for better performance.

4.4.2 Intersecting-stitch lengths

Similar to the orientation process, we use the reaction-diffusion equation

$$\frac{\mathrm{d}\xi}{\mathrm{d}t}=(\xi^*-\xi)+\lambda_\xi\sum_n\omega_n(\xi_n-\xi)+\varepsilon_\xi \quad (5)$$

for the length case, where the local reaction term and local diffusion term are aperiodic this time. In reality, the lengths of embroidery threads that are stitched into the embroidery base cloth are restricted. Namely, threads that are too long are easy to break and that are too short are difficult to stitch into the cloth. For this reason, we pre-define a global range of available stitch length $\xi \in [\xi_{\min}, \xi_{\max}]$. In view of real embroidery, we set $\xi_{\min}$=5 mm and $\xi_{\max}$=50 mm. Generally, the stitch length used in a more salient object or edge (e.g., fur, petal, and pistil) is smaller than that in a simple background object or less salient region (e.g., sky, mountain, and river). There have been many salient edge, ridge, or region detection methods (Guo *et al.*, 2007), which usually generate a grayscale image (essentially a 2D matrix with the saliency value falling in the range of 0–1 or 0–255). It is the same size as the source image. However, the embroidery cloth is usually larger than the image and possesses different measurement criteria. Hence, inspired by Huang *et al.* (2011), we first prepare a grayscale image which is called an importance map, then divide the cloth into a set of rectangular grids by cutting repeatedly along x and y directions via the importance map, and finally obtain a non-uniform partition (Fig. 6). The larger the importance value is, the smaller the grid area is. Notice that a different importance map can lead to a different partition of grids. Finally, the external length force ξ^* can be acquired using

$$\xi^* = \xi_{\min} + (\xi_{\max} - \xi_{\min}) S^* / S_{\max}, \tag{6}$$

where S^* is the area of the grid in which the current intersecting-stitch is located, and $S_{\max}$ is the maximum area of all grids in the current region.

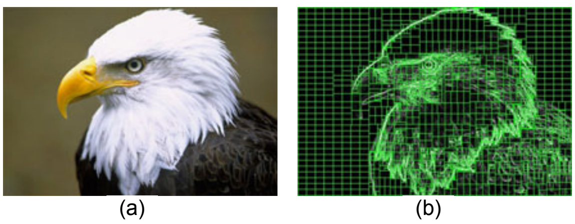

Fig. 6 Image non-uniform grids partition
(a) Source image; (b) Non-uniform grids

The diffusion factor λ_ξ controls the local length contrast of neighboring intersecting-stitches. A larger λ_ξ leads to more locally similar stitch lengths, and thus lower length contrast among neighboring intersecting-stitches. On the contrary, a higher contrast shows the layout where long and short intersecting-stitches are alternately distributed. Correspondingly, the Markov field energy to be optimized is

$$E(\xi) = \sum_i (\xi_i - \xi_i^*)^2 + \lambda_\xi \sum_i \sum_{j \in N(i)} \omega_{ij} (\xi_i - \xi_j)^2. \tag{7}$$

4.4.3 Intersecting-stitch colors

Color contrast (sometimes called tempo by artists) is an intuitive yet powerful tool to depict styles, especially in oil painting. As is known to us, the RNE artists have drawn lessons from western oil painting theories, and actually many RNE artworks also exhibit hue contrast, light-shade contrast, or smooth color transition among neighboring stitches. That is also an important factor different from the traditional embroidery. We apply three reaction-diffusion processes for three channels (hue, saturation, and lightness) in the HSV space, respectively. Similar to θ^* and ξ^*, the reference colors $c^*=(h^*, s^*, v^*)$ are obtained from the source image. Here are three reaction equations:

$$\frac{\mathrm{d}h}{\mathrm{d}t} = \sin(h^* - h) + \lambda_h \sum_n \omega_n \sin(h_n - h) + \varepsilon_h, \tag{8}$$

$$\frac{\mathrm{d}s}{\mathrm{d}t} = (s^* - s) + \lambda_s \sum_n \omega_n (s_n - s) + \varepsilon_s, \tag{9}$$

$$\frac{\mathrm{d}v}{\mathrm{d}t} = (v^* - v) + \lambda_v \sum_n \omega_n (v_n - v) + \varepsilon_v. \tag{10}$$

Their corresponding Markov field energy equations are

$$E(h) = \sum_i \varphi(h_i - h_i^*) + \lambda_h \sum_i \sum_{j \in N(i)} \omega_{ij} \varphi(h_i - h_j), \tag{11}$$

$$E(s) = \sum_i (s_i - s_i^*)^2 + \lambda_s \sum_i \sum_{j \in N(i)} \omega_{ij} (s_i - s_j)^2, \tag{12}$$

$$E(v) = \sum_i (v_i - v_i^*)^2 + \lambda_v \sum_i \sum_{j \in N(i)} \omega_{ij} (v_i - v_j)^2. \tag{13}$$

4.5 Stitch rendering

After all the stitches are updated, we can obtain a collection of stitches whose two endpoints' coordinates in the embroidery cloth space and RGB colors are easy to calculate. Next, we create the RNE rendering image by drawing these stitches on canvas. In this process, we adopt a simple strategy to simulate a single stitch with texture information (Fig. 7). The algorithm is as follows:

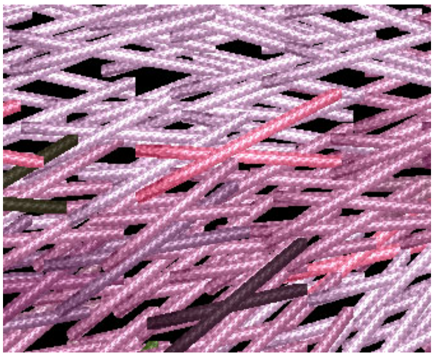

Fig. 7 Stitch rendering detail

1. Map current stitch into the corresponding canvas coordinates by the ratio of its cloth coordinates and the cloth size. Calculate the start pixel and the end pixel in the canvas of the current stitch.

2. Interpolate all pixels between the start pixel and end pixel using the Bresenham algorithm. Extend this one pixel width line to the d pixel width line via the thread image width.

3. Transform the stitch color and thread image color from RGB to YUV; calculate the ratio of lightness of each pixel in the thread image and average lightness.

4. Weaken the lightness at the endpoint of the current thread in the Y channel by

$$Y(x_i, y_i) = (0.75 + 0.25 d_i / d_0) Y(x_i, y_i), \tag{14}$$

where d_i is the distance of the current pixel to the center pixel of the thread and d_0 is the threshold and we set its value as 45% of the current thread length.

5. Render the current thread into the canvas.

5 Results

The proposed method is implemented using Microsoft Visual Studio 2010 in such configuration: Intel Core 2 Duo E8400 3.0 GHz CPU, 2 GB memory, Windows XP. It usually takes approximately 30–50 s to update the stitch layout of one region for the single reaction-diffusion process. Fortunately, the diffusion algorithm can be parallelized. Available parallel programming API (e.g., OpenMP for C++) combined with multi-core processors or graphics hardware can speed up this process for real-time applications.

5.1 Convergence

The proposed method uses several reaction-diffusion procedures to process stitch attributes for a stitch layout based on a stitch neighborhood graph. After a few diffusion iterations, the algorithm tends to convergence and all attributes of stitches tend to a stable state. This means the Markov energy and attributes variations tend to convergence. Take stitch orientation diffusion as an example. The Markov energy and the average value of orientation variations (small variation means stability, and large variation means instability) at each iteration are calculated. As Fig. 8 shows, we plot both energy and variation values for positive diffusion (λ>0) and negative diffusion (λ<0), respectively. The variation value at each iteration is the average orientation change value $\Delta\theta$.

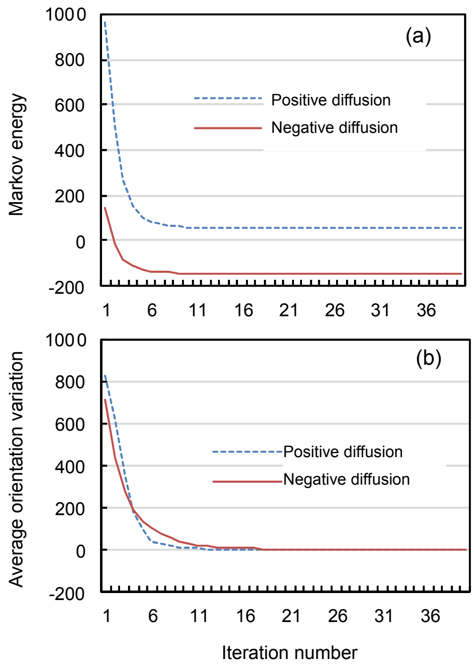

Fig. 8 Orientation diffusion of Markov energy iterations (a) and average orientation variation iterations (b)

Notice that the convergent Markov energy is positive when λ>0 and negative when λ<0 because it is essentially a statistic value of the attribute differences between all the stitches and their neighbors. Even so, the energy value decreases until convergence and the final state tends to stability. The average orientation variation results further prove it.

5.2 Layout parameters efficiency

Different diffusion factors λ's control different attributes, and thus affect different stitch layout styles. The following results show the effect of λ on orientations, lengths, and colors, respectively.

5.2.1 Stitch orientations

Fig. 9 shows the capability of positive diffusion forcing the neighboring stitches' orientations to be random on purpose. Fig. 10 shows the opposite case, where the negative diffusion forces the neighboring stitches' orientations to be chaotic.

As a quantitative confirmation, we do statistical analysis of the orientations and plot the histogram of differences in orientations between all neighboring stitches, where the horizontal axis represents orientation differences and the vertical axis represents the ratio of the current difference. Figs. 11a and 11b show the initial and final histograms for Figs. 9 and 10, respectively.

A histogram with a higher peak and lower tails means that neighboring stitches have similar orientations and thus most difference values should be close to zero. On the contrary, a histogram that is nearly flat or has a lower peak and heavier tails shows the randomness or low similarity among stitch orientations. Fig. 12 shows results (Fig. 12a–12c) and histograms for different λ_θ (Fig. 12d). Obviously, the larger the λ_θ is, the sharper its histogram. This shows the efficiency and flexibility of our stitch orientation layout control.

5.2.2 Stitch lengths

Similar work is done for stitch lengths (Fig. 13). Larger λ_ξ leads to similar lengths between neighboring stitches and smaller λ_ξ leads to a higher number of different lengths. Users can control the overall layout of stitch lengths using an appropriate λ_ξ.

5.2.3 Stitch colors

Our approach can create vibrant colors and high contrast colors via hue diffusion, saturation diffusion, and lightness diffusion. Figs. 14a–14c show the flower results of different λ_h's; Figs. 14e–14g are their corresponding hue distributions. It is observed that smaller λ_h creates new colors and makes the whole object more vivid. In fact, many artists like using high contrast colors to emphasize the object, even though these colors do not exist in the source image. The histograms with different λ_h's (similar to the orientation case, see Fig. 14d) further illustrate the hue contrasts. Fig. 14h shows the corresponding energy iterations for Fig. 14d. The lower the energy value is, the more hues it owns. Figs. 15 and 16 show saturation diffusion and lightness diffusion, respectively.

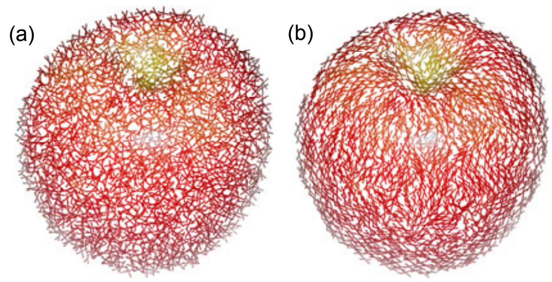

Fig. 9 Positive diffusion effect (λ_θ=0.9) forcing initial chaotic orientations of stitches (a) to be similar in the final stable stitches (b)

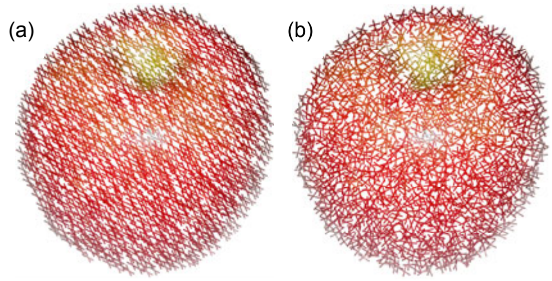

Fig. 10 Negative diffusion effect (λ_θ=−0.9) forcing initial similar or regular orientations of stitches (a) to be chaotic in final chaotic stitches (b)

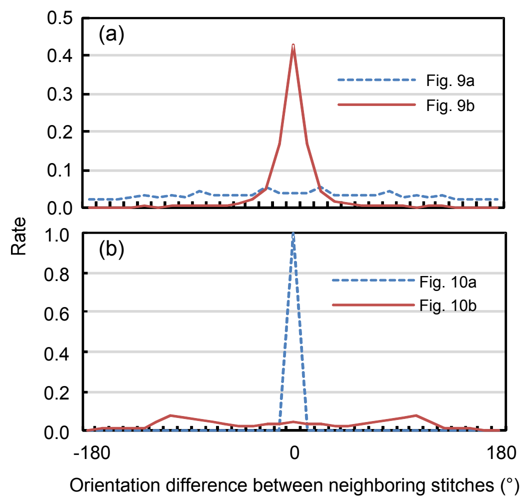

Fig. 11 Histogram of orientation differences between neighboring stitches

(a) Positive diffusion; (b) Negative diffusion. Both the initial (Figs. 9a and 10a) and final (Figs. 9b and 10b) histograms are plotted

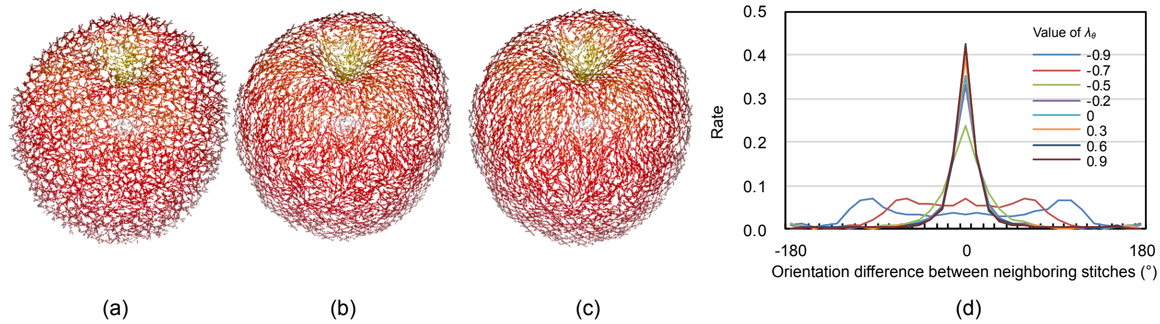

Fig. 12 Effects of λ_θ on stitch orientation

(a) λ_θ=−0.9; (b) λ_θ=−0.5; (c) λ_θ=0.6; (d) Histograms of different λ_θ's

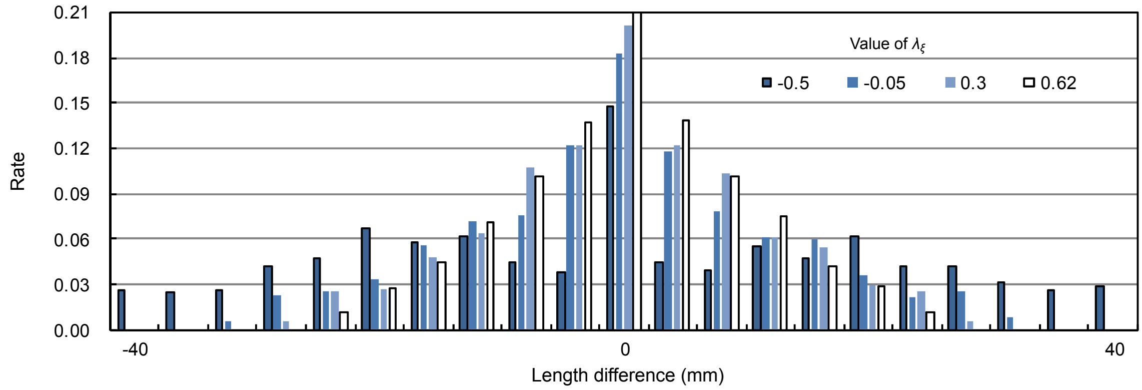

Fig. 13 Effects of λ_ξ on stitch length

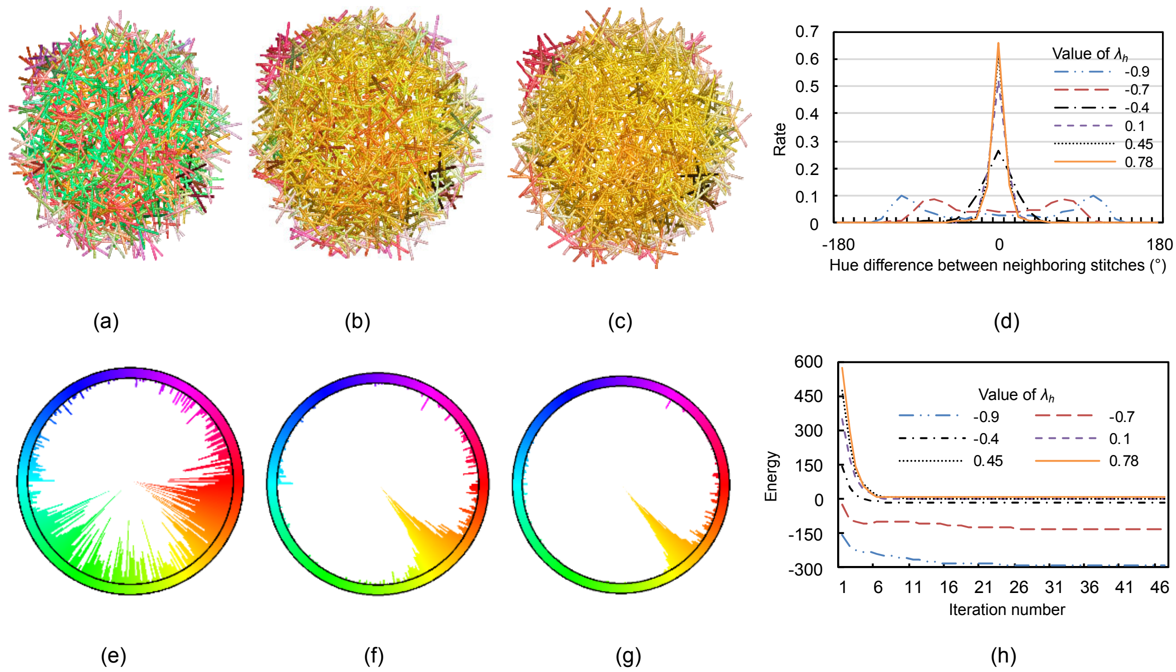

Fig. 14 Effects of λ_h on stitch hue

(a) λ_h=−0.9; (b) λ_h=0.1; (c) λ_h=0.78; (d) Histograms of different λ_h's; (e) λ_h=−0.9; (f) λ_h=0.1; (g) λ_h=0.78; (h) Energy iterations of different λ_h's

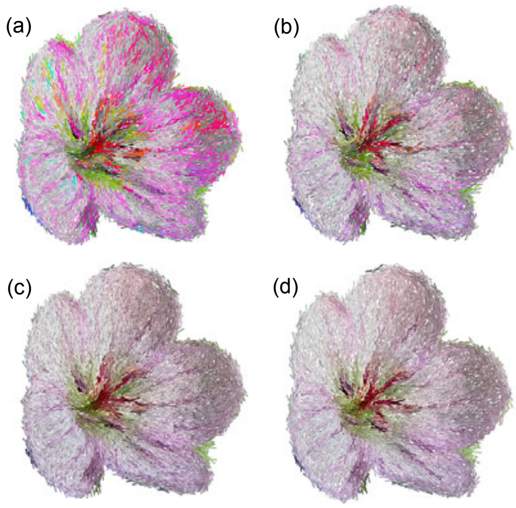

Fig. 15 Effects of λ_s on stitch saturation
(a) λ_s=−0.9; (b) λ_s=−0.6; (c) λ_s=0.35; (d) λ_s=0.64

Fig. 17 Four examples of user-drawn guide curves and results

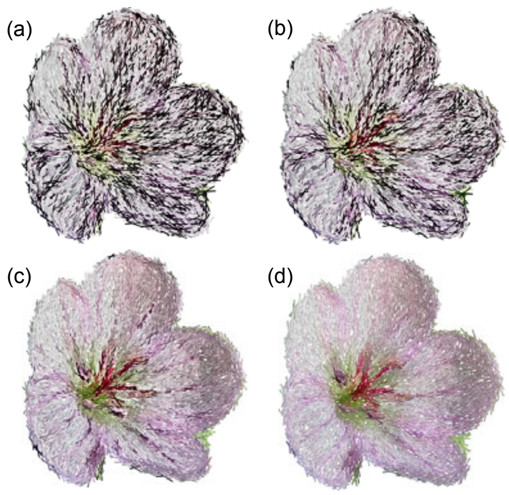

Fig. 16 Effects of λ_v on stitch lightness
(a) λ_v=−0.9; (b) λ_v=−0.6; (c) λ_v=0.35; (d) λ_v=0.64

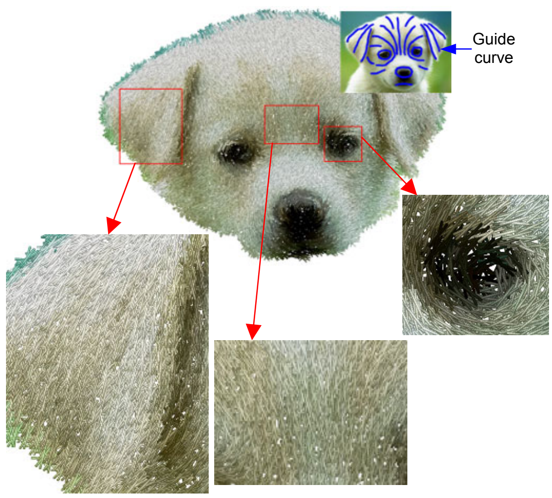

Fig. 18 A user interaction example of guide curves and final results with local details

5.3 User interaction

The proposed approach also supports users in guiding the stitch orientation layout interactively, considering the fact that no automatic orientation field method works well for any object and meanwhile the users sometimes expect to create their own desired layout result. Fig. 17 shows four examples: users draw different curves and obtain the corresponding diffusion results. We can see that the final layout style obeys the user's intent and the stitches follow the curves' directions. A more complex example is shown in Fig. 18, in which several local results marked by rectangles are cut out to show the details.

5.4 User study

The five high-level layout parameters (diffusion factors) have a conspicuous influence on stitch layout styles. We choose several classes of labelled objects from the Microsoft Research Cambridge Object Recognition Image Database (Shotton *et al.*, 2006) and our image database. Then we ask 20 participants to select their most desirable parameter values for each object by repeated adjustment. Finally, we collect all user-selected parameter values of all objects and just give the general value range of each parameter for each class (Table 1). Note that we do not attempt to compute the absolute parameter values for different objects or evaluate the qualities of user-selected results since our start point in this paper is to supply users with a controllable stitch layout tool for RNE. Fig. 19 includes some final rendering examples.

6 Conclusions

This paper presents a controllable stitch layout strategy for RNE. To express different stitch styles, we design a stitch layout model including low-level stitch attributes, low-level intersecting-stitch attributes, and especially high-level layout parameters that mainly control the overall layout rule of different stitches attributes (orientations, lengths, and colors). For this purpose, a topological structure named a stitch neighborhood graph is built to model the interactions among neighboring stitches and several reaction-diffusion procedures are adopted to process different stitches attributes respectively based on the high-level layout parameters and stitch neighborhood graph. Users can easily change the stitch layout styles by adjusting high-level layout parameters or drawing several guide curves interactively. Experimental results show that our approach is capable of reflecting various stitch layout styles and is flexible for user interaction.

There are, however, several drawbacks to this method. Generally, it works well and exhibits better controllability for those images containing clear objects. However, it is not suitable for some scenery images for the reason that too complicated color or texture variations cannot be expressed using a single model. Besides, our high-level layout parameters are not user intuitive. More automatic and powerful stitch layout methods still need to be studied.

References

Adabala, N., Magnenat-Thalmann, N., Fei, G., 2003. Real-time rendering of woven clothes. Proc. ACM Symp. on Virtual Reality Software and Technology, p.41-47. [doi:10.1145/1008653.1008663]

Bratkova, M., Shirley, P., Thompson, W.B., 2009. Artistic rendering of mountainous terrain. *ACM Trans. Graph.*, **28**(4):1-17. [doi:10.1145/1559755.1559759]

Chen, H., Zhu, S.C., 2006. A generative sketch model for human hair analysis and synthesis. *IEEE Trans. Patt. Anal. Mach. Intell.*, **28**(7):1025-1040. [doi:10.1109/TPAMI.2006.131]

Chen, S.G., Sun, Z.X., Xiang, J.H., *et al.*, 2011. Research on the technology of computer aided irregular needling embroidery. *Chin. J. Comput.*, **34**(3):526-538 (in Chinese). [doi:10.3724/SP.J.1016.2011.00526]

Chen, X.L., McCool, M., Kitamoto, A., *et al.*, 2012. Embroidery modeling and rendering. Proc. Graphics Interface Conf., p.131-139.

Table 1 General ranges of five layout parameters for RNE

Object	λ_θ	λ_ξ	λ_h	λ_s	λ_v
Animal	0.16–0.52	−0.12–0.43	0.05–0.90	0.45–0.87	−0.05–0.15
Bird	0.02–0.53	−0.07–0.39	0.1–0.6	−0.03–0.71	0.01–0.15
Grass	−0.85–0.50	0.10–0.86	0.8–0.9	0.50–0.86	0.04–0.50
Flower	−0.9–−0.7 or 0.50–0.78	0.40–0.89	−0.87–−0.30 or 0.05–0.50	−0.1–0.3	0.10–0.67
Sky	0.8–0.9	−0.83–−0.41	0.7–0.9	0.00–0.34	−0.4–0.4

λ_θ, λ_ξ, λ_h, λ_s, and λ_v are diffusion factors for orientation, length, hue, saturation, and lightness, respectively


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

Fig. 19 Final results of our proposed approach for RNE works of Apple (a), Flower (b), Bird (c), and Dog (d)

Collomosse, J.P., Hall, P.M., 2002. Painterly rendering using image salience. Proc. 20th Eurographics UK Conf., p.122-128. [doi:10.1109/EGUK.2002.1011281]

Curtis, C.J., Anderson, S.E., Seims, J.E., *et al.*, 1997. Computer-generated watercolor. Proc. 24th Annual Conf. on Computer Graphics and Interactive Techniques, p.421-430. [doi:10.1145/258734.258896]

Deussen, O., Hiller, S., van Overveld, C., *et al.*, 2000. Floating points: a method for computing stipple drawings. *Comput. Graph. Forum*, **19**(3):41-50. [doi:10.1111/1467-8659.00396]

Ding, Z.A., Zhang, X., Chen, W., *et al.*, 2012. Coherent streamline generation for 2-D vector fields. *Tsinghua Sci. Technol.*, **17**(4):463-470. [doi:10.1109/TST.2012.6297592]

Dogrusoz, U., Belviranli, M.E., Dilek, A., 2013. CiSE: a circular spring embedder layout algorithm. *IEEE Trans. Visual. Comput. Graph.*, **19**(6):953-966. [doi:10.1109/TVCG.2012.178]

Gamito, M.N., Maddock, S.C., 2009. Accurate multidimensional Poisson-disk sampling. *ACM Trans. Graph.*, **29**(1): 8:1-8:19. [doi:10.1145/1640443.1640451]

Gansner, E.R., Hu, Y.F., North, S., 2013. A maxent-stress model for graph layout. *IEEE Trans. Visual. Comput. Graph.*, **19**(6):927-940. [doi:10.1109/TVCG.2012.299]

Guo, C.E., Zhu, S.C., Wu, Y.N., 2007. Primal sketch: integrating structure and texture. *Comput. Vis. Image Underst.*, **106**(1):5-19. [doi:10.1016/j.cviu.2005.09.004]

Hausner, A., 2001. Simulating decorative mosaics. Proc. 28th Annual Conf. on Computer Graphics and Interactive Techniques, p.573-580. [doi:10.1145/383259.383327]

Hays, J., Essa, I., 2004. Image and video based painterly animation. Proc. 3rd Int. Symp. on Non-photorealistic Animation and Rendering, p.113-120. [doi:10.1145/987657.987676]

Hertzmann, A., 2003. A survey of stroke-based rendering. *IEEE Comput. Graph. Appl.*, **23**(4):70-81. [doi:10.1109/MCG.2003.1210867]

Huang, H., Fu, T.N., Li, C.F., 2011. Painterly rendering with content-dependent natural paint strokes. *Vis. Comput.*, **27**(9):861-871. [doi:10.1007/s00371-011-0596-5]

Inglis, T.C., Inglis, S., Kaplan, C.S., 2012. Op art rendering with lines and curves. *Comput. Graph.*, **36**(6):607-621. [doi:10.1016/j.cag.2012.03.003]

Kaldor, J.M., James, D.L., Marschner, S., 2008. Simulating kintted cloth at the yarn level. *ACM Trans. Graph.*, **27**(3): 65:1-65:9. [doi:10.1145/1360612.1360664]

Kalnins, R.D., Markosian, L., Meier, B.J., *et al.*, 2002. WYSIWYG NPR: drawing strokes directly on 3D models. *ACM Trans. Graph.*, **21**(3):755-762. [doi:10.1145/566654.566648]

Kang, H., Lee, S.Y., Chui, C.K., 2007. Coherent line drawing. Proc. 5th Int. Symp. on Non-photorealistic Animation and Rendering, p.43-50. [doi:10.1145/1274871.1274878]

Kondo, S., Miura, T., 2010. Reaction-diffusion model as a framework for understanding biological pattern formation. *Science*, **329**(5999):1616-1620. [doi:10.1126/science.1179047]

Kopf, J., Cohen-Or, D., Deussen, O., *et al.*, 2006. Recursive Wang tiles for real-time blue noise. *ACM Trans. Graph.*, **25**(3):509-518. [doi:10.1145/1141911.1141916]

Litwinowicz, P., 1997. Processing images and video for an impressionist effect. Proc. 24th Annual Conf. on Computer Graphics and Interactive Techniques, p.407-414.

Lu, J., Sander, P.V., Finkelstein, A., 2010. Interactive painterly stylization of images, videos and 3D animations. Proc. ACM SIGGRAPH Symp. on Interactive 3D Graphics and Games, p.127-134. [doi:10.1145/1730804.1730825]

Luft, T., Deussen, O., 2006. Real-time watercolor illustrations of plants using a blurred depth test. Proc. 4th Int. Symp. on Non-photorealistic Animation and Rendering, p.11-20. [doi:10.1145/1124728.1124732]

Mao, X.Y., Nagasaka, Y., Imamiya, A., 2002. Automatic generation of pencil drawing from 2D images using line integral convolution. SIGGRAPH, p.149. [doi:10.1145/1242073.1242162]

Martín, D., Arroyo, G., Luzón, M., *et al.*, 2010. Example-based stippling using a scale-dependent grayscale process. Proc. 8th Int. Symp. on Non-photorealistic Animation and Rendering, p.51-61. [doi:10.1145/1809939.1809946]

O'Donovan, P., Hertzmann, A., 2012. AniPaint: interactive painterly animation from video. *IEEE Trans. Visual. Comput. Graph.*, **18**(3):475-487. [doi:10.1109/TVCG.2011.51]

Park, Y., Yoon, K., 2008. Painterly animation using motion maps. *Graph. Models*, **70**(1):1-15. [doi:10.1016/j.gmod.2007.06.001]

Peng, C.H., Yang, Y.L., Wonka, P., 2014. Computing layouts with deformable templates. *ACM Trans. Graph.*, **33**(4), Article 99. [doi:10.1145/2601097.2601164]

Perona, P., 1998. Orientation diffusions. *IEEE Trans. Image Process.*, **7**(3):457-467. [doi:10.1109/83.661195]

Salisbury, M.P., Anderson, S.E., Barzel, R., *et al.*, 1994. Interactive pen-and-ink illustration. Proc. 21st Annual Conf. on Computer Graphics and Interactive Techniques, p.101-108. [doi:10.1145/192161.192185]

Sanderson, A.R., Kirby, R.M., Johnson, C.R., *et al.*, 2006. Advanced reaction-diffusion models for texture synthesis. *Graph. GPU Game Tools*, **11**(3):47-71. [doi:10.1080/2151237X.2006.10129222]

Santella, A., DeCarlo, D., 2002. Abstracted painterly renderings using eye-tracking data. Proc. 2nd Int. Symp. on Non-photorealistic Animation and Rendering. [doi:10.1145/508530.508544]

Secord, A., 2002. Weighted Voronoi stippling. Proc. 2nd Int. Symp. on Non-photorealistic Animation and Rendering, p.37-43. [doi:10.1145/508530.508537]

Shiraishi, M., Yamaguchi, Y., 2000. An algorithm for automatic painterly rendering based on local source image approximation. Proc. 1st Int. Symp. on Non-photorealistic Animation and Rendering, p.53-58. [doi:10.1145/340916.340923]

Shotton, J., Winn, J., Rother, C., *et al.*, 2006. Textonboost: joint appearance, shape and context modeling for multi-class object recognition and segmentation. Proc. 9th European Conf. on Computer Vision, p.1-15. [doi:10.1007/11744023_1]

Turk, G., 1991. Generating textures on arbitrary surfaces using reaction-diffusion. Proc. 18th Annual Conf. on Computer Graphics and Interactive Techniques, p.289-298. [doi:10.1145/122718.122749]

Vanderhaeghe, D., Barla, P., Thollot, J., *et al.*, 2007. Dynamic point distribution for stroke-based rendering. Proc. 18th Eurographics Conf. on Rendering Techniques, p.139-146. [doi:10.2312/EGWR/EGSR07/139-146]

Wyvill, B., Overveld, K., Carpendale, S., 2004. Rendering cracks in batik. Proc. 3rd Int. Symp. on Non-photorealistic Animation and Rendering, p.61-149. [doi:10.1145/987657.987667]

Xie, N., Laga, H., Saito, S., *et al.*, 2010. IR2s: interactive real photo to Sumi-e. Proc. 8th Int. Symp. on Non-photorealistic Animation and Rendering, p.63-71. [doi:10.1145/1809939.1809947]

Xu, J., Kaplan, C.S., 2007. Image-guided maze construction. *ACM Trans. Graph.*, **26**(3):29-38. [doi:10.1145/1276377.1276414]

Xu, K., Wang, J.P., Tong, X., *et al.*, 2009a. Edit propagation on bidirectional texture functions. *Comput. Graph. Forum*, **28**(7):1871-1877. [doi:10.1111/j.1467-8659.2009.01565.x]

Xu, K., Li, Y., Ju, T., *et al.*, 2009b. Efficient affinity-based edit propagation using K-D tree. *ACM Trans. Graph.*, **28**(5), Article 118. [doi:10.1145/1661412.1618464]

Xu, K., Chen, K., Fu, H.B., *et al.*, 2013. Sketch2scene: sketch-based co-retrieval and co-placement of 3D models. *ACM Trans. Graph.*, **32**(4):123:1-123:12. [doi:10.1145/2461912.2461968]

Xu, X., Zhang, L., Wong, T.T., 2010. Structure-based ASCII art. *ACM Trans. Graph.*, **29**(4), Article 52. [doi:10.1145/1778765.1778789]

Yamamoto, S., Mao, X., Imamiya, A., 2004. Colored pencil filter with custom colors. Proc. 12th Pacific Conf. on Computer Graphics and Applications, p.329-338. [doi:10.1109/PCCGA.2004.1348364]

Yang, Y.L., Wang, J., Vouga, E., *et al*., 2013. Urban pattern: layout design by hierarchical domain splitting. *ACM Trans. Graph*, **32**(6):181:1-181:12. [doi:10.1145/2508363.2508405]

Yuan, X.R., Che, L.M., Hu, Y.F., *et al.*, 2012. Intelligent graph layout using many users' input. *IEEE Trans. Visual. Comput. Graph.*, **18**(12):2699-2708. [doi:10.1109/TVCG.2012.236]

Zeng, K., Zhao, M.T., Xiong, C.M., *et al*., 2009. From image parsing to painterly rendering. *ACM Trans. Graph.*, **29**(1):2:1-2:11. [doi:10.1145/1640443.1640445]