Patchy: An Interactive Patchwork Design System

Yuki Igarashi*  
University of Tsukuba / Meiji University

Jun Mitani  
University of Tsukuba

1 Introduction

Patchwork is a well-known type of needlework that involves sewing pieces of fabric into a larger design. It is commonly used to form quilts, but can also be used to make bags, wall hangings, cushion covers, and other items. Larger designs are usually based on repeating patterns, which are built up using different shapes. Professional patchwork designers design original patterns; however, novice users usually use geometric patterns or off-the-shelf patterns for each piece; this is because it is difficult for novices to design patterns while visualizing the resulting larger fabric.

Here we propose an interactive system to assist the design of original patchwork patterns. Coahranm and Fiume [2005] presented a sketch-based design system for a specific quilting artform, i.e., Bargello patterns. They described an algorithm that transforms sketched input data into graceful Bargello curves. In contrast, with our system the user designs original patchwork strokes. The user can design original patchwork patterns using various fabric colors through a process of trial and error.

2 User Interface

Figure 1 shows an overview of the process. The user first designs patchwork pattern silhouettes using a painting interface (Fig. 1a). Then s/he designs fabric patterns through trial and error, as shown in Figure 1(b). The user draws two types of stroke: a silhouette stroke, which is the border of the pieces of fabric, and a stitch stroke, which is a stitch line on the pieces of fabric. The system automatically adds stitch strokes, as shown in Figure 1(c). The user can also design stitch strokes using a free-form stroke (Fig. 1d), and then sew the patchwork, as shown in Figure 1(g).

As shown in Figure 1(e,f), the user can grab a stroke and pull it to deform the shape of the stitch lines using the peeling interface introduced by Igarashi et al. [2005]. When s/he drags and drops a texture image into a domain bounded by silhouette strokes, the system updates the region inside this area. The user can also examine the overall color balance in a distant view using the zoom tool.

3 Implementation

Our prototype system was implemented in Java™{}, and demonstrated in real-time on a 2.1-GHz Core™{}i7 PC. The system uses a hybrid approach combining vector and raster representations. The user’s strokes are stored as vector graphic primitives, and the system renders the strokes and texture as a raster image after computing the normals.

The system computes the normals of the cells after the user inputs or modifies a stroke, as shown in Figure 2. The target normal $n$ of a cell $v(x, y)$ is computed as the normal of a height field defined by

$$h(d) = \begin{cases} \sqrt{r^2 - (r - d)^2} & (d < r) \\ \frac{r}{d} & (d \geq r) \end{cases},$$

where $d$ is the distance to the closest stroke and $r$ is a constant parameter; we used $r = 30$. Then the system applies smoothing to each normal 50 times.

4 Results

We used our system to create patchwork designs, and then the resulting pattern was manually sewn, as shown in Figure 3. Each design session typically required 10 – 15 min, and the sewing required 3 – 5 h. Users can quickly experiment with various patterns using the system before beginning sewing.

References
