Embroidery Modeling and Rendering in Real Time

Xinling Chen[*]
University of Waterloo

Michael McCool[†]
Intel/University of Waterloo

Asanobu Kitamoto[‡]
National Institute of Informatics, Japan

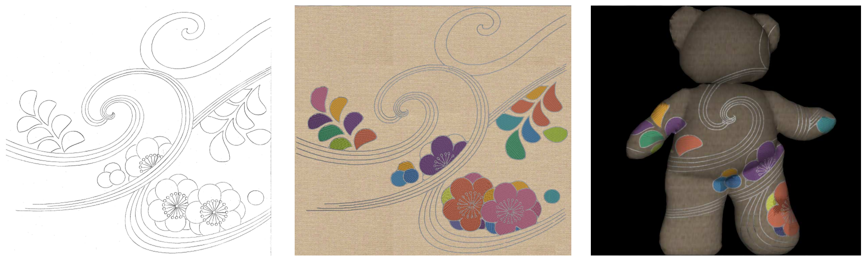

Figure 1: *The extraction phase analyses a line drawing to determine regions and lines. The modeling phase places stitches according to traditional embroidery techniques. The rendering phase dynamically generates a lighting-dependent and antialiased texture which is mapped onto a potentially deforming 3D object.*

1 Introduction

Many non-photorealistic rendering algorithms have been developed recently that simulate different traditional artistic styles including painting, mosaics, and stippling. However, there is at least one more traditional approach to rendering images non-photorealistically: embroidery. Embroidery creates images by stitching threads of different colours into a base cloth. In free hand embroidery, many different stitch styles are possible and stitches can be placed relatively freely on the surface.

There is no previous work dealing specifically with stitch placement to simulate embroidery, but there are some applicable techniques. For example, based on Lloyd's Method, a common iterative technique to place samples by relaxation, Hausner [Hausner 2001] developed an algorithm to distribute small square tiles to form synthetic mosaics. The stippling algorithm presented by Secord successfully produces attractive results in a natural way by using a weighted variant of Lloyd's method [Secord 2002].

Embroidery also has a set of traditional patterns. As a reference, we have been studying the book *Traditional Japanese Patterns* [Kurenai-Kai 2005] which defines a set of sewing techniques used in Japanese free embroidery.

The goal of our work is to transform a traditional embroidery pattern (typically consisting mainly of region outlines) into a high-quality embroidery image generated in real time. Real time rendering enables the user to navigate around the image and manipulate it. We also want to be able to place the embroidery on deforming objects and use dynamic lighting models for the threads.

2 Our Approach

We achieved embroidery simulation and rendering in three phases. In the first phase, vision algorithms were applied to extract lines and regions in embroidery patterns scanned from traditional sources. In the second phase, the selection and placement of primitives in the identified regions is performed. This is a semi-automatic process since some artistic choice is involved, including but not limited to stitching style, thread parameters and colors. Therefore, an interactive interface was built. We developed an algorithm to simulate a classic boundary style, *line of staggered diagonals*, as well as an algorithm to simulate the most used and oldest region-filling stitch to be found in embroidery, *long-short stitching*. Since artists often put the stitches along the longest axis of the region, we compute the main inertia axis of each segmented region [Hiller et al. 2003] and use this to orient the stitching. Moreover, we separated lines that represent details from the region outline into a second layer. This was done manually although this could possibly be achieved automatically in future work using vision algorithms. When we render the model, we can easily place these details above the fill layer, which is similar to what real embroiderer would normally do for such details. In the third phase, stitches are dynamically lit in real time in surface space by a simple thread thread lighting model [Mallo et al. 2005], but modulated with alpha mapping and tangent modifications to simulate curved stitches. Our rendering technique supports high-quality real-time rendering on deformable objects using hardware acceleration. Before rendering, we compute the tangents from the texture parameterization [Lengyel 2001]. Then light and view vectors as well as local surface frame are captured and projected onto the local frame to light the stitches. Lastly, a filtered texture pyramid, which supports good antialiasing, is constructed from the resulting texture and applied to the 3D object.

[*]e-mail: cherrychen0602@gmail.com
[†]e-mail: michael.mccool@intel.com
[‡]e-mail: kitamoto@nii.ac.jp

References

HAUSNER, A. 2001. Simulating decorative mosaics. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '01, 573–580.

HILLER, S., HELLWIG, H., AND DEUSSEN, O. 2003. Beyond stippling – methods for distributing objects on the plane. *Eurographics 22*, 515–522.

KURENAI-KAI. 2005. *Traditional Japanese Patterns 1*. Seigensha.

LENGYEL, E. 2001. *Computing Tangent Space Basis Vectors for an Arbitrary Mesh*. Terathon Software 3D Graphics Library, http://www.terathon.com/code/tangent.html.

MALLO, O., PEIKERT, R., SIGG, C., AND SADLO, F. 2005. Illuminated lines revisited. In *IEEE Visualization Conference*, 19–26.

SECORD, A. 2002. Weighted Voronoi stippling. In *Proceedings of the 2nd international symposium on Non-photorealistic animation and rendering*, ACM, NPAR '02, 37–43.

SIGGRAPH 2011, Vancouver, British Columbia, Canada, August 7 – 11, 2011.
ISBN 978-1-4503-0921-9/11/0008