Designing Look-And-Feel Using Generalized Crosshatching

Yuxiao Du
Department of Visualization, Texas A&M University
College Station, Texas 77831

Ergun Akleman
Departments of Visualization & Computer Science and Engineering, Texas A&M University
College Station, Texas 77831
ergun.akleman@gmail.com

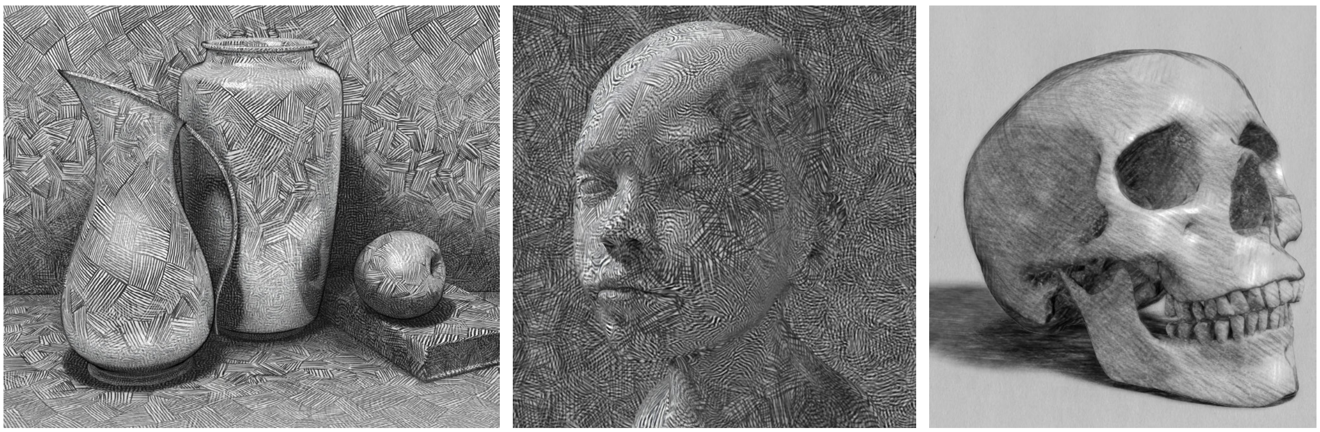

Figure 1: *Examples of results from our generalize cross-hatching approach. As it can be seen from these examples, we can obtain consistent look-and-feel of hand-drawn charcoal drawing effect for a variety of materials and shapes.*

ABSTRACT

In this work, we have developed an approach to include any crosshatching technique into any rendering system with global illumination effects (see Figure 1). Our new approach provide a robust computation to obtain hand-drawn effects for a wide variety of diffuse and specular materials. Our contributions can be summarized as follows: (1) A Barycentric shader that can provide generalized crosshatching with multi-textures; and (2) A texture synthesis method that can automatically produce crosshatching textures from any given image.

CCS CONCEPTS

• **Computing methodologies → Non-photorealistic rendering**;

KEYWORDS

ACM proceedings, LaTeX, text tagging

ACM Reference format:
Yuxiao Du and Ergun Akleman . 2017. Designing Look-And-Feel Using Generalized Crosshatching. In *Proceedings of SIGGRAPH '17 Talks, Los Angeles, CA, USA, July 30 - August 03, 2017,* 2 pages.
https://doi.org/10.1145/3084363.3085054

SIGGRAPH '17 Talks, July 30 - August 03, 2017, Los Angeles, CA, USA

ACM ISBN 978-1-4503-5008-2/17/07…$15.00
https://doi.org/10.1145/3084363.3085054

1 INTRODUCTION AND MOTIVATION

During the last two decades many non-photorealistic (NPR) rendering methods have been developed to emulate existing crosshatching techniques [Curtis et al. 1997; Winkenbach and Salesin 1994]. However, there has been no formal approach that can provide artists freedom to develop their own crosshatching style that has never been done before. This is mainly because NPR is not usually viewed as emulation of existing artistic techniques. On the other hand, concept of NPR can be viewed as an artistic device by itself to provide generalized methods to artists for the development of distinct artistic style in their artworks. In this work, we present a new approach that can be used by artists to develop their own crosshatching styles.

Development of a rendering and shading framework to obtain a desired look-and-feel is an important task for any artistic application from movie making to game development. Shade Trees architecture laid the foundations of the procedural shader concept to create any desired look-and-feel [Cook 1984] . Despite the success of shaders and shading languages [Hanrahan and Lawson 1990], to obtain a desired style is still a challenge even for highly qualified lighting technical directors.

2 RELATED WORK

Recently, a new approach, called Barycentric Shaders, is developed to simplify shader development through a more intuitive and streamlined process [Akleman et al. 2016]. In Barycentric shaders, the shading equations are given in the form of $C(u, v) = \sum_{i=0}^{M} B_i(t)T_i(u, v)$ where $C(u, v)$ is rendered color of the point (u, v), $T_i(u, v)$'s are texture images (we call control images) and t is one of the shading parameters such as diffuse or specular parameters, ambient occlusion or shadow and $B_i(t)$'s are basis functions that satisfy partition of the unity. Using such basis functions guarantees that regardless of how the shading parameters are computed during rendering, we can always obtain a consistent style [Akleman et al. 2016]. The key decisions, therefore, are to obtain desired styles are the choices of basis functions and control images.

These shaders can easily be used in any standard rendering pipeline along with global illumination as a streamlined process. Although, Barycentric shaders have successfully used in emulating existing artists' styles [Du and Akleman 2016; Liu and Akleman 2015], they have never been used to create non-existing styles. In this work, we present a more general application of Barycentric Shaders that can be used to obtain any style that could roughly be classified as crosshatching. Our goal in this work is to obtain generalized crosshatching effects in a streamlined process that can be used in a standard rendering pipeline.

We, therefore, used a specific Barycentric shader introduced to obtain charcoal effect [Du and Akleman 2016]. This shader provides an appropriate mixture of textures for styles that requires sharp transitions such as crosshatching, charcoal or cartoon. The heart of sharp transitions comes from using degree zero B-spline in Barycentric formulation. This set of basis functions guarantees that if a parameter u is computed between $u_i \leq u < u_{i+1}$, we choose the color from control image $T_i(u, v)$. If the average color of control image $T_i(u, v)$ is between between u_i and u_{i+1}, this guarantees to obtain a clean texture that provides approximately desired color value.

3 METHODOLOGY AND IMPLEMENTATION

The key part of our approach is a texture synthesis algorithm that can construct on average constant color crosshatching textures from a single crosshatching image that can be hand-drawn by artists (See Figure 2). The original image can be any image that is turned into a repeating wallpaper image. Using our algorithm, we first turn the original image into an approximately constant color image. We then obtain darker and lighter shaded versions as shown in Figure 2. These textures are later mapped to objects by using tri-planar projection. This projection is simple and partly avoids foreshortening caused by perspective on texture. There may be other texture mapping strategies that may provide better results, but we did not see any visible problem in wide range of shapes.

Our texture synthesis algorithm is an extremely simple non-stationary convolution filter that consists of two stages as follows: (1) For every pixel, compute the average color in a region around the pixel where the shape of the region and weights are controlled by a user-defined kernel. (2) If the color of the region is darker than the goal color, make the color of the pixel lighter; or otherwise darker.

In our actual implementation, we apply dilation filter to make region darker and erosion filter to make region lighter. For dilation and erosion, we use maximum and minimum since they can work on gray-scale images. Therefore, even if the original images are gray-scale, our algorithm still works as shown in Figure 2. We usually use gray-scale crosshatching to avoid aliasing.

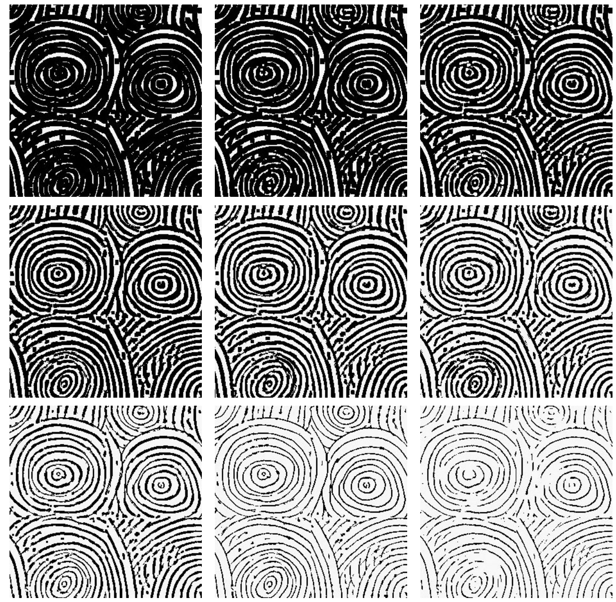

Figure 2: *Details of textures that are obtained using our image synthesis algorithm that produced lighter or darker textures that resemble an original crosshatching texture.Algorithm works not only for B&W images, but also gray-scale and color images.*

4 CONCLUSION AND FUTURE WORK

This algorithm also works for color images by applying the same texture synthesis algorithm to the three channels of the original image instead of one. Although, we have not truly investigated this direction, based on these initial results, this approach also seems to have potential for obtaining wider variety of styles.

This work was partially supported by NVidia through a hardware donation and the National Science Foundation under Grants NSF-EFRI-1240483, NSF-CMMI-1548243 and NSF-ECCS-1547075.

REFERENCES

E. Akleman, S. Liu, and D. House. 2016. Barycentric Shaders: Art Directed Shading Using Control Images. In *Proceedings of Expresive'2016.* Eurographics Association, Lisbon, Portugal, 39–49.

Robert L Cook. 1984. Shade trees. *ACM Siggraph Computer Graphics* 18, 3 (1984), 223–231.

Cassidy J Curtis, Sean E Anderson, Joshua E Seims, Kurt W Fleischer, and David H Salesin. 1997. Computer-generated watercolor. *Proceedings of ACM SIGGRAPH Computer Graphics* 31, 4 (1997), 421–430.

Yuxiao Du and Ergun Akleman. 2016. Charcoal Rendering and Shading with Reflections. In *ACM SIGGRAPH 2016 Posters (SIGGRAPH '16).* ACM, Pasadena, CA, USA, 32:1–32:2.

Pat Hanrahan and Jim Lawson. 1990. A language for shading and lighting calculations. *Proceedings of ACM SIGGRAPH Computer Graphics* 24, 4 (1990), 289–298.

Siran Liu and Ergun Akleman. 2015. Chinese ink and brush painting with reflections. In *SIGGRAPH 2015: Studio.* ACM, Los Angeles, CS, USA, 8.

Georges Winkenbach and David H Salesin. 1994. Computer-generated pen-and-ink illustration. *Proceedings of ACM SIGGRAPH Computer Graphics* 28, 4 (1994), 91–100.