Frayed Cell Diagrams

Michael Burch, Corinna Vehlow, and Daniel Weiskopf*
Visualization Research Center (VISUS), University of Stuttgart

Abstract

Tesselation-based or area-based visual representations are common to many artistic or visualization applications. For example, Voronoi art uses a space-filling tessellation of the image by Voronoi cells. We present frayed cell diagrams as an aesthetic visual representation of the separating border between those space-filling regions. Our approach is based on a simple randomized algorithm that densely draws lines toward the reference points of cells. This algorithm is controlled by a few parameters whose effects are detailed in the paper: the density and size of cells, the degree of fraying, and the color coding. To demonstrate the usefulness of frayed diagrams for algorithmic art, we applied them to pieces of Voronoi art. Finally, we conducted a survey to assess the aesthetics of the frayed cell diagrams. As a result, we found out that the majority of the participants preferred a high degree of fraying, but that a non-negligible subgroup preferred diagrams without any fraying.

CR Categories: I.3.3 [Computer Graphics]: Picture/Image Generation

Keywords: Edge rendering, algorithmic art, Voronoi art.

1 Introduction

Many diagrammatic representations make use of space-filling area-based visual representations, i.e., they rely on a tessellation of the canvas. Tessellation plays an important role in art, ranging from highly symmetric mosaic tilings that range back to Sumerian and later to Roman times [Field 1993], to Moorish ornaments [Grünbaum et al. 1986] and quilting for motives [Porter 2006], and all the way to the graphic art by Escher [Escher 1992]. Tessellation has strong connections to the mathematics of geometry, but, at the same time, provides aesthetic visual representations frequently used in art. Therefore, it is a compelling meeting point of mathematics and art [Sarhangi 1992, Ch. 5]. A particularly interesting example of tessellation comes with Voronoi diagrams because it combines mathematical-geometric aspects with algorithms and art, prominently conveyed in pieces of Voronoi art [Kaplan 1999].

In this paper, we address the aesthetic visual representation of the *borders* between the areas that build the tessellation. When it comes to several neighboring areas, those implicitly and naturally share a common border. Typically, this border is represented by a "hard" and equally thick line separating the neighboring areas from each other. We argue that we can improve the aesthetics of area representations by improving the design of the separating graphical elements. A compelling visual style is highly relevant for applications in algorithmic art because it allows us to design an overall visual "look".

Our contribution is a rendering technique that leads to *frayed boundaries* of the areas of a tessellation or tiling. We define a border as being frayed if there is a randomized structure in the thickness property of the border. We introduce a simple randomized algorithm that can generate frayed borders as a means for producing algorithmic art. Our algorithm is designed to work with a *cell*-based decomposition of the image; this decomposition does not necessarily need to cover the full image space, but may contain holes. However, we assume that the boundaries of the areas enclose cells that can be associated with a reference point within that cell. The strength of the algorithm is its simplicity and the fact that even corners (where two or more borders meet) are treated consistently and without any visual artifacts. Our algorithm allows easy-to-control changes of the governing parameters, in particular, its degree of fraying. To illustrate the usefulness of our algorithm, we explore the space of control parameters by showing their effects on the visual appearance. In particular, we demonstrate that fraying diagrams fit very well to Voronoi art. Finally, we assess the aesthetics of frayed cell diagrams by a survey.

2 Related Work

As discussed in the introduction, tessellation has been, and keeps on, playing a relevant role in many areas of art and visual design. An example that is particularly closely related to our paper is Voronoi art: this can be regarded as algorithmic art that leads to aesthetic imagery [Kaplan 1999]. Other examples of Voronoi art include portraits constructed as Voronoi diagrams [Levin 2000]. However, such previous work typically focuses on the construction of the tiling and its artistic effect, but not on advanced rendering styles for the boundary curves that separate the Voronoi cells. Although we also use examples of Voronoi art in this paper, the Voronoi aspect is not a novel contribution of our work. In fact, we rather use Voronoi art to demonstrate how our new frayed cell diagrams can be effectively used for that application example.

In this paper, we investigate the problem of making borders in area-based representations more distinguishable by applying the concept of frayed diagrams. Instead of using complex rendering techniques, our approach is based on a simple randomized algorithm based on line rendering. Our rendering technique shares some aspects of non-photorealistic pen-and-ink rendering that typically mimics traditional stroke-based rendering [Hertzmann 2003; Deussen and Isenberg 2013], including stippling and hatching techniques. However, even when tonal art maps [Praun et al. 2001] or other hatching techniques were applied to rendering "thick" lines, frayed rendering would be technically difficult because we want to achieve consistent fraying across corners of cells and also a random distribution of the jags that allow for quite wide fraying. There also exist visualizations that use frayed borders, generated by attaching lines perpendicular to the border of map elements [Kim et al. 2013; Isenberg 2013]: the lines can be used to visually encode multivariate attributes [Kim et al. 2013] or add stylistic shading to produce aesthetic renditions [Isenberg 2013].

Other algorithmic art approaches that yield images visually related to ours include stained glass windows [Mould 2003], random

*E-mail: {michael.burch,corinna.vehlow,daniel.weiskopf}@visus.uni-stuttgart.de

CAe 2014, August 08 – 10, 2014, Vancouver, British Columbia, Canada.

ACM 978-1-4503-3019-0/14/08 $15.00

trees [Burch and Weiskopf 2013], and dentric stylization [Long and Mould 2009]. However, none of these papers addresses the problem of rendering frayed boundaries.

3 Rendering Algorithm

This section describes a new rendering method to generate frayed cell diagrams. The key aspect of this algorithm is that it visits all points (pixels) of the image and potentially draws lines from the pixel to a reference point within the cell—in a randomized fashion.

Algorithm 1 shows the pseudo code of the algorithm. The algorithm takes the image canvas I and the fraying parameter δ as input. Then, it iterates over all pixels p. For each one of these pixel positions, the algorithm may render a straight line to the reference point of the cell that contains that pixel. The randomized nature of the algorithm is introduced here: depending on the value of a random variable (drawn from a uniform distribution), the line is drawn or skipped. The critical part of the algorithm is the subroutine $getReferencePoint(p)$: it returns the reference point of the cell; if the pixel is outside any cell or on the border of a cell, the subroutine returns a NULL value. In other words, this subroutine essentially provides a membership test for the pixel position p because it has to determine which cell contains this pixel. Based on that information, the cell's reference point is returned.

Algorithm 1 Rendering frayed cell diagram

FrayedCellRendering(I, δ):

I; // Image consisting of pixels to be filled
δ; // Fraying factor between 0 and 1

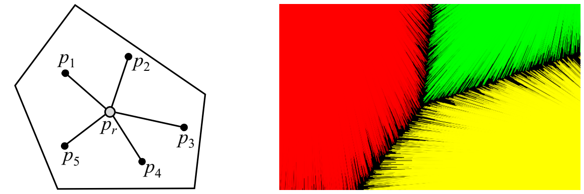
r; // Random value between 0 and 1
p_r; // Reference point within a cell

// Iterate over all pixels:
for all $p \in I$ **do**
  $p_r := getReferencePoint(p)$;
  // if within a cell:
  if $p_r \neq$ NULL **then**
    $r := random()$;
    if $r \geq \delta$ **then**
      $drawLine(p, p_r)$;
    end if
  end if
end for

The reference point may be any point within the cell, as long as a straight line from any other point within the cell to the reference is completely contained within the cell. For example, this is the case for any point within a convex cell. However, even some non-convex cells may have such reference points, i.e., star-shaped polygons.

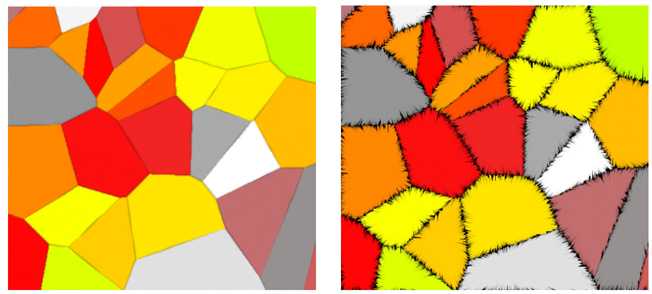
Figure 1(a) illustrates the rendering process for a simple case that contains just one convex cell. Here, only a few randomly selected lines are drawn toward the reference point. In practice, the image will be densely covered by lines.

For aesthetic images, we recommend placing the reference point somewhere in the "middle" of the cell. In this way, isotropy of the rendering patterns can be achieved. For example, this point could be the center of mass within a convex cell. Another approach is to start with a set of points and then build an implicit Voronoi diagram from those points. In fact, the images of this paper are generated in this way. Figure 2 shows an example of such a diagram. Figure 2(a) depicts the traditional Voronoi diagram, which can be rendered by using the membership function $getReferencePoint(p)$ and coloring points according to the cell index (or reference point); the



(a) (b)

Figure 1: *Frayed diagrams: (a) illustration of the rendering process, (b) consistent direction of the fraying pattern at a cell corner.*



(a) (b)

Figure 2: *(a) Non-frayed and (b) frayed Voronoi diagram.*

membership function just has to determine the closest input point for the pixel p. In comparison, Figure 2(b) shows the frayed cell diagram for the same point set.

One advantage of the algorithm is its obvious simplicity. In fact, the pseudo code of Algorithm 1 is a faithful representation of the length of a typical implementation of our algorithm. Another advantage is that the patterns are consistent along the frayed boundary, even at sharp corners. Figure 1(b) shows a typical case. Here, three cells meet at a common vertex. Within each cell, the "spikes" of the boundary point toward the same point and, thus, do not show any inconsistency or rapid change of the direction of the strokes. In contrast, it would be hard to achieve such consistency by applying tonal art maps or other textures along edges of the cells because the underlying quadrilaterals would change direction in a discontinuous fashion at the vertices.

The third advantage of our algorithm is that it provides a convenient way of rendering a frayed pattern from a geometric distribution. Again, this would be hard with any texturing approach for polylines because the geometric distribution may lead to patterns that could reach very far into the cell. To understand this effect, the frayed boundary can be interpreted as the result of a Bernoulli process from statistics. For this interpretation, let us consider a 1D case in which the boundary is the pixel at the origin and the reference point is along the x axis at infinity. Then, for each pixel p visited by Algorithm 1, we have a chance of δ that this pixel does not trigger the rendering of a line. To have a fraying pattern of length k, we need to have k subsequent cases of such non-rendering, followed by one trial that triggers the rendering process. The probability for this scenario is: $\delta^k(1-\delta)$. The corresponding distribution is a geometric distribution. The expected value of a geometrically distributed random variable is:

$$E = \sum_{k=0}^{\infty} \delta^k (1-\delta) \cdot k = \frac{\delta}{1-\delta} \ .$$

The above observation is true for any line between a cell boundary and the reference point. Therefore, the average thickness of the frayed boundary is $\delta/(1-\delta)$ pixels wide. Put differently, we have a direct control of the width of the frayed boundary in the form of the parameter δ.

4 Control of Parameters

In this section, we illustrate how parameter changes influence the appearance of frayed cell diagrams.

Density The size of cells and their density have a major influence on the appearance of the frayed cell diagrams. Compared to Figure 2, where the cells have random size, Figure 3 shows a decrease of cell size from left to right. A side effect of changing cell density is the change of apparent brightness. With a constant fraying factor, all boundaries have equal width. Therefore, the right part of Figure 3 (with smaller cells) looks darker than the left part (with larger cells and, thus, fewer frayed boundaries).

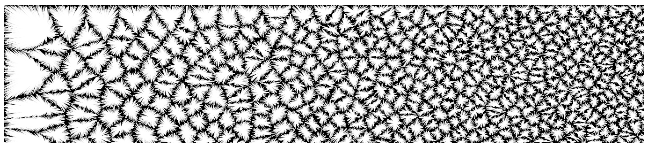


Figure 3: *Change of cell density: the size of the convex polygons decreases from left to right (fraying factor δ = 0.85).*

Degree of Fraying Besides the size of cells, the second major influence on the appearance is the degree of fraying described by the factor δ. As discussed above, δ provides a direct control of the average line width. The possible values for δ range from 0 to 1. Figure 4 illustrates the visual effect of changing the fraying factor. For $\delta = 0$, we obtain the traditional diagram with hard borders (since $getReferencePoint(p)$ yields NULL in Algorithm 1).

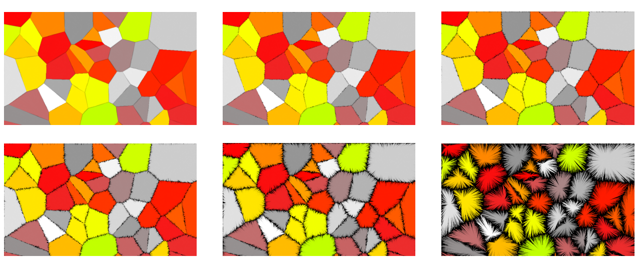


Figure 4: *Effect of changing the fraying factor δ = 0.0, 0.2, 0.4, 0.6, 0.8, 0.95 (left-to-right).*

Color Mapping Finally, we can apply a color map when we render the lines in Algorithm 1. By default, we render white lines on black background, leading to black frayed boundaries around white cells. However, by adapting the rendering color, we can produce cells of varying color. In Figures 2 and 4, we randomly assigned a color from a color map to each cell. However, other color maps and more advanced color assignment would be possible as well (e.g., driven by outside information; or varying even within a cell).

5 Application to Voronoi Art

We apply frayed cell diagrams to Voronoi art. Traditionally, Voronoi art uses hard boundaries to separate the Voronoi cells. However, we believe that frayed edges add to the aesthetics of Voronoi art because they introduce (slightly randomized) visual patterns along the cell boundaries.

Technically, a finite set of points in the Euclidean plane builds the basis for our Voronoi illustrations. For Figures 2 and 4, this set of points was generated randomly. For our Voronoi art examples, we extract the reference points from an image. In particular, we use the Floyd-Steinberg dithering approach [Floyd and Steinberg 1976] with subsequent jittering to produce reference points on a regular grid of the image plane. Through this process, we implicitly control the density, size, and position of the cells in the frayed diagram. Therefore, it can be interpreted as one way of controlling the density parameter.

Figures 5 and 6 show examples of Voronoi art. Figure 6 also illustrates the effect of the fraying factor. Please note that the background of the underlying image is completely white. Therefore, there are no reference points outside the bunny, leading to this skeleton-like embedding of the bunny.

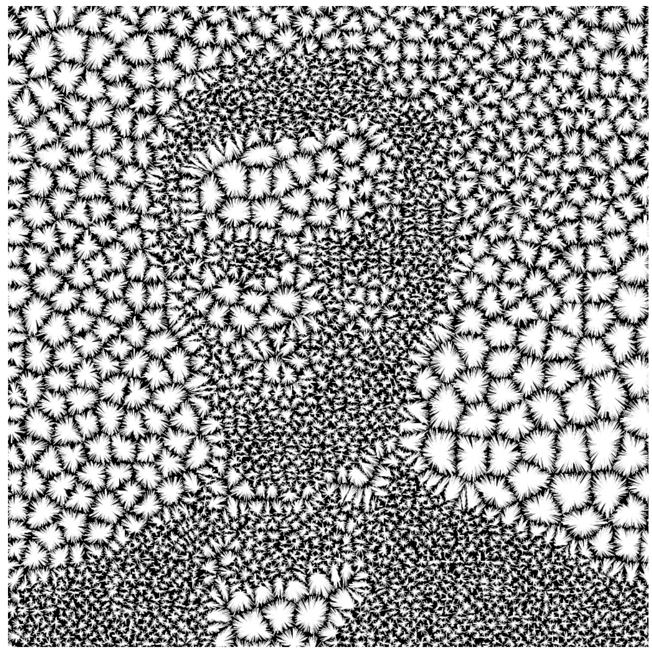


Figure 5: *Example of Voronoi art in the form of a frayed cell diagram (fraying factor δ = 0.85).*

6 Aesthetics Survey

We conducted an electronic survey to assess the aesthetics of frayed cell diagrams. In particular, we were interested in finding out whether there are certain preferred values for the fraying factor. The survey contained 11 figures of the same Voronoi diagram, with a different fraying factor $\delta \in \{0.0, 0.1, 0.2, ..., 0.9, 0.95\}$. Figure 4 shows a few of the images that were used for the survey. There were 45 participants that volunteered from our institute. The participants were instructed to name the figure that they found most aesthetically appealing.

The results of our survey showed that the majority of the participants preferred frayed representations over non-frayed traditional ones (see the histogram in Figure 7). Interestingly, the histogram shows a clear separation of the participants in two groups: the minority prefers the traditional line rendering, whereas the majority prefers frayed diagrams with a rather large fraying factor.

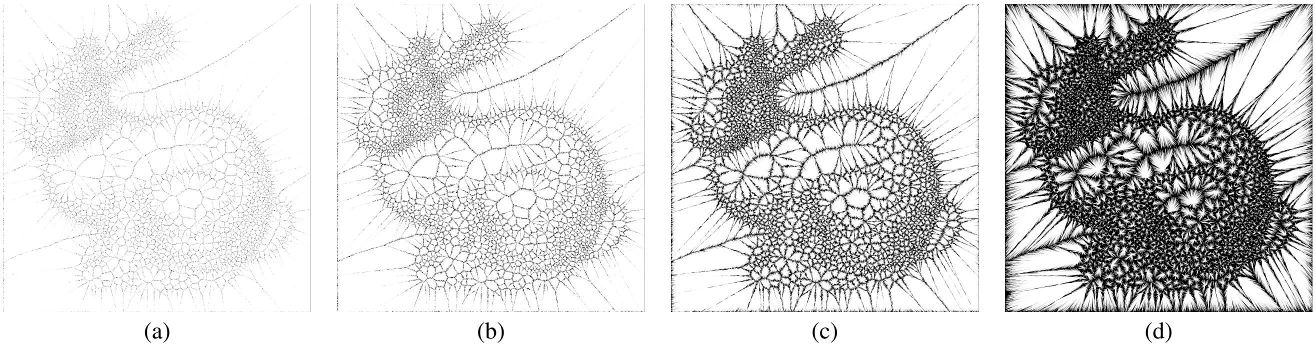


Figure 6: *Effect of changing fraying parameter: (a) Frayed cell diagram with $\delta = 0.2$. (b) 0.5 fraying factor. (c) 0.8 fraying factor. (d) 0.9 fraying factor. Please note that the algorithm is able to produce aesthetically looking, consistent frayed lines even at the meeting points of the border lines.*

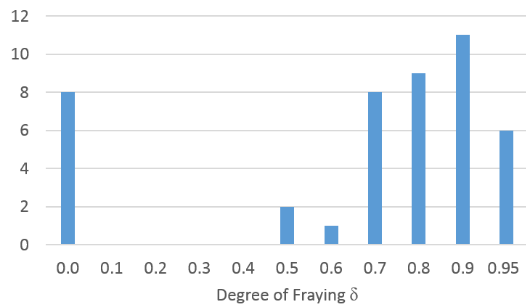


Figure 7: *Histogram of answers to our aesthetics survey showing how many people voted for which fraying factor δ.*

7 Conclusion and Future Work

We have investigated the concept of frayed cell diagrams. We have shown how they can be rendered with a simple randomized algorithm that draws lines from pixels to the reference point in the respective cell. The only assumption is that the lines have to be completely within the cell. Our algorithm is governed by three easy-to-control parameters: the density and size of the cells, the fraying factor, and the color map. We have demonstrated the effects of those parameters for several typical settings. Furthermore, we have applied frayed cell diagrams to pieces of Voronoi art generated from gray-scale input images. From our survey, we found out that there seem to be two groups of people—one of which prefers traditional line renderings and the other one prefers frayed boundaries; and that participants with preference for frayed diagrams vote for rather strong fraying.

So far, we have applied frayed cell rendering to Voronoi diagrams. Other shapes of tilings could be explored in future applications for art production. We also see a good potential that frayed cell rendering could be applied to diagrams from computer-based visualization and statistical graphics; for example, Bristle Maps [Kim et al. 2013] could be adopted to combine frayed diagrams with visualization techniques.

References

BURCH, M., AND WEISKOPF, D. 2013. The aesthetics of rapidly-exploring random trees. In *Proceedings of the Symposium on Computational Aesthetics*, 45–52.

DEUSSEN, O., AND ISENBERG, T. 2013. Halftoning and stippling. In *Image and Video-Based Artistic Stylisation*, P. Rosin and J. Collomosse, Eds. Springer, 45–61.

ESCHER, M. C. 1992. *M. C. Escher: The Graphic Work*. Taschen.

FIELD, R. 1993. *Geometric Patterns from Roman Mosaics: And How to Draw Them*. Tarquin.

FLOYD, R., AND STEINBERG, L. 1976. An adaptive algorithm for spatial grey scale. *Proceedings of the Society of Information Display 17*, 75–77.

GRÜNBAUM, B., GRÜNBAUM, Z., AND SHEPHARD, G. C. 1986. Symmetry in Moorish and other ornaments. *Computers & Mathematics with Applications 12*, 3-4, 641–653.

HERTZMANN, A. 2003. A survey of stroke-based rendering. *IEEE Computer Graphics and Applications 23*, 4, 70–81.

ISENBERG, T. 2013. Visual abstraction and stylisation of maps. *The Cartographic Journal 50*, 1, 8–18.

KAPLAN, C. S. 1999. Voronoi diagrams and ornamental design. In *Proceedings of the First Annual Symposium of the International Society for the Arts, Mathematics, and Architecture (ISAMA '99)*, 277–283.

KIM, S., MACIEJEWSKI, R., MALIK, A., JANG, Y., EBERT, D., AND ISENBERG, T. 2013. Bristle Maps: A multivariate abstraction technique for geovisualization. *IEEE Transactions on Visualization and Computer Graphics 19*, 9, 1438–1454.

LEVIN, G., 2000. Segmentation and symptom. `http://www.flong.com/projects/zoo`.

LONG, J., AND MOULD, D. 2009. Dendritic stylization. *The Visual Computer 25*, 3, 241–253.

MOULD, D. 2003. A stained glass image filter. In *Proceedings of the Eurographics Workshop on Rendering*, 20–25.

PORTER, C. 2006. *Tessellation Quilts: Sensational Designs From Interlocking Patterns*. David & Charles.

PRAUN, E., HOPPE, H., WEBB, M., AND FINKELSTEIN, A. 2001. Real-time hatching. In *Proceedings of the ACM SIGGRAPH Conference*, 581–586.

SARHANGI, R. 1992. *Elements of Geometry for Teachers*. Pearson Custom Publishing.