EG UK Computer Graphics & Visual Computing (2020)
P. D. Ritsos and K. Xu (Editors)

# Continuous-Line-Based Halftoning with Pinwheel Tiles

Abdalla G. M. Ahmed†

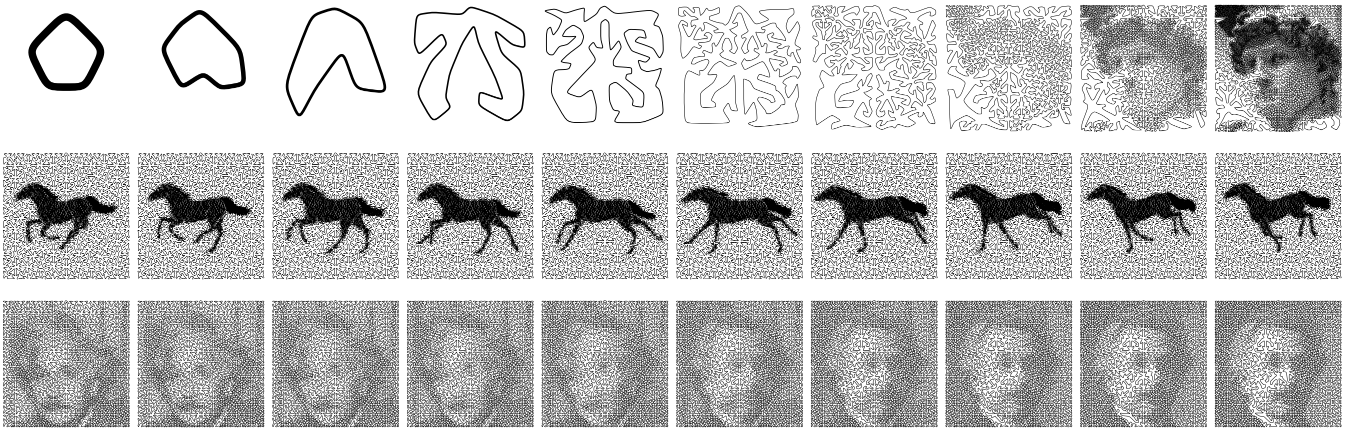

**Figure 1:** *Various kinds of animations enabled by our continuous-line drawing framework. The example in the top is suitable for transition effects in presentations, the one in the middle suits animated GIFs, while the morphing one in the bottom demonstrates that we can even cope with arbitrary motion pictures.*

**Abstract**

*We describe a technique for continuous-line-based halftoning with space-filling curves, namely Segerman's pinwheel curve, which is based on Conway's pinwheel tiles. We adapt the length of the curve passing through the tiles in accordance with a supplied density map, e.g. a grayscale image. We use the subdivision depth of the tiling for a coarse control of the curve length, and we describe a finer control to faithfully match the density map. Our technique exhibits excellent temporal coherence, making it suitable for animated images.*

## 1. Introduction

Halftoning is the process of rendering a continuous-tone image using a discrete set of tones, typically only one. It was originally meant to render full-tone images in monochrome devices such as printers, and the goal is to make a distant view of the rendered image indistinguishable from the continuous-tone image. This is achieved by controlling the density of the constituent elements, e.g. dots of ink, so that it matches the level of grayness in the corresponding part of the given image [PB96]. A close-up of the halftoned image, however, would inevitably reveal the discretized nature of the halftoning process, and the constituent elements become visible.

Halftoning is an old process, mainly associated with printing. Digital halftoning emerged with the advent of digital computers, and received a lot of attention in the seventies and eighties of last century. The primary goal was to conceal the constituent color elements, taking the nature of the printer into consideration. This is known as dithering [Uli87]. The generally accepted principle is that an even distribution of the elements with no directional preference (isotropic) achieves the best dithering results. Ulichney [Uli87, Uli88] coined the name "blue noise" to describe such a distribution, characterizing it by its frequency power spectrum.

In contrast to the main stream in halftoning, aimed at concealing the constituent elements that produce the tone of the image, a different line of interest emerged aimed at stylized halftoning, where a close-up reveals some interesting or unexpected details. For example, Ostromoukhov [OH95] used text and artistic elements, and Bosch [Bos11] used duotone Truchet tiles as elements.

With the advent of high-resolution printing technology, the ba-

† Email:abdalla_gafar@hotmail.com





DOI: 10.2312/cgvc.20201149

sic problem of halftoning has become less important. On the other hand, the need for stylized halftoning continues to evolve in the "Facebook" era, where personalizing images is highly desirable. In this paper we present a new approach for stylized halftoning, based on the pinwheel tiling [Rad94]. The concept we present bear a distinctive style, as in Figure 1, is fast, enabling interactive or even real-time rates, and exhibits excellent temporal coherence, enabling the stylization of animations or morphing.

## 2. Related Work

Line-based halftoning is a sub-category of stylized halftoning where the elements are straight or curved line segments. The key idea is to use a curve-generating process, and parameterize it such that it generates longer scribbled curves at darker parts of the rendered image. The style varies significantly with the employed technique. For example, Pnueli and Bruckstein [PB96] develop Schroeder idea [Sch83] of employing Eikonal equation, which produces nested non-intersecting contour lines, Pedersen and Singh [PS06] use an iterative process to develop a labyrinth-like continuous curve, Browne [Bro08] uses Truchet-like contours, leading to a bumped curve, Ahmed [Ahm14] uses a recursive subdivision of the rendered image, leading to different axes-aligned styles, and Ahmed and Deussen [AD16] use an ensemble of amplitude-modulated waves to render scan lines of the image.

A generic approach for producing density-controlled lines is to connect density-controlled sample points. This idea was first introduced by Bosch and Herman [BH04], who propose solving the traveling salesman problem (TSP) to connect the points. The idea was greatly improved by Kaplan and Bosch [KB05], who renamed it TSP Art. An equally-impressive style was presented by Inoue and Urahama [IU09], who propose using a minimum spanning tree (MST) to connect the points, which is easier to compute than TSP. Ahmed [Ahm15] describes a generic tone-correction approach to account for moving from dimension-less points to one-dimensional lines, and describes a few more styles. The sample points are typically generated as a blue noise distribution using, for example, the blue-noise optimal transport (BNOT) algorithm [dGBOD12]. Thanks to this blue-noise distribution of the underlying points, all these styles are highly isotropic. TSP Art is distinctively characterized by a non-self-intersecting flow of the closed curve. On the downside, these methods call for a preliminary process like BNOT for distributing the points, and another process for solving the designated connectivity model, which adds significant complexity. The method we present below is self-contained and simple, and produces a non-self-intersecting close curve as well.

Recently, Stoppel and Bruckner [SB19] introduced an integrated framework for stylized halftoning. Instead of defining a specific style, their framework is plugin-based, enabling the users to mix and match between different styles, and define their own new styles. In contrast to this, our method is self-contained, producing the whole halftoning curve as a single continuous line.

Recently, Swart [Swa17] described an approach to morph TSP Art. In addition to the inherent complexity of computing TSP Art, there is extra computation of the morphing. In contrast, method we describe here exhibits excellent temporal coherence, and readily enables animation at interactive or even real-time rates.

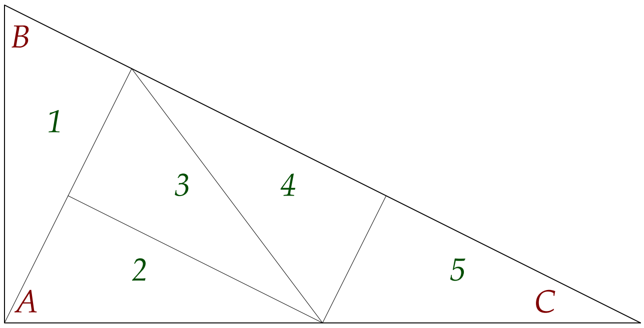


**Figure 2:** *A pinwheel tile, and its subdivision into five child tiles.*

## 3. Rendering with Pinwheel Curve

Pinwheel tiles are right-angled triangles of side lengths 1 and 2, that can be subdivided into 5 smaller copies of themselves, as illustrated in Figure 2. They are attributed to the late John Conway, and they received special interest of established mathematicians like Conway and Radin [Rad94] because they are the first known rep-tile (i.e self-paving tile), among only a few, that keeps adding new tile orientations as the subdivision process is continued.

Akin to space-filling curves like Peano's and Hilbert's, Segerman [Seg14] described a space-filling curve based on pinwheel tiling, as in Figure 3, obtained by connecting the centroids of triangles in a specific order, as illustrated in Figure 4. The idea is quite simple, but care must be taken in ordering the triangles to ensure a non-self-intersecting curve. This can be worked out by inspection. Observing the labeling in Figure 2, let the curve pass from vertex B to C, i.e from 1 to 5, as in the leftmost triangle in Figure 4. Then we observe that the flow is inverted to go from C to B in the middlemost child triangles 3 and 4. Thus, a continuous curve is obtained by: a) processing the child tiles in the designated order, either 1-to-5 or 5-to-1, and b) designating an inverted order of the parent's for child 3 and 4. We provide a sample code in the supplemental materials.

The idea for using pinwheel curves for halftoning is simple: we subdivide more at darker parts of the underlying image, producing a squigglier, hence longer, hence darker curve. Realizing this idea, is not trivial, though, as we will see through the following discussion. While other space-filling curves can be used, the distribution of centroids of pinwheel tiles (Figure 5) inherently exhibits a blue noise spectrum, as noted by Ahmed [Ahm19], which is highly desirable, as mentioned in the introduction.

### 3.1. Subdivision

We start by designating a line weight to indicate the nominal amount of darkness per unit length of our curve. This is a parameter availed to the end users, through which they control the level of detail they would like to render. We then initialize a stack with the four top-most triangles in Figure 4, and initialize a list of output points with the top-center point (0.5, 1). The structure to describe a tile contains two elements: a transformation matrix $m$ that maps the relative coordinates inside the tile to absolute coordinates, and a Boolean parameter $BtoC$ telling whether the tile is traversed from B

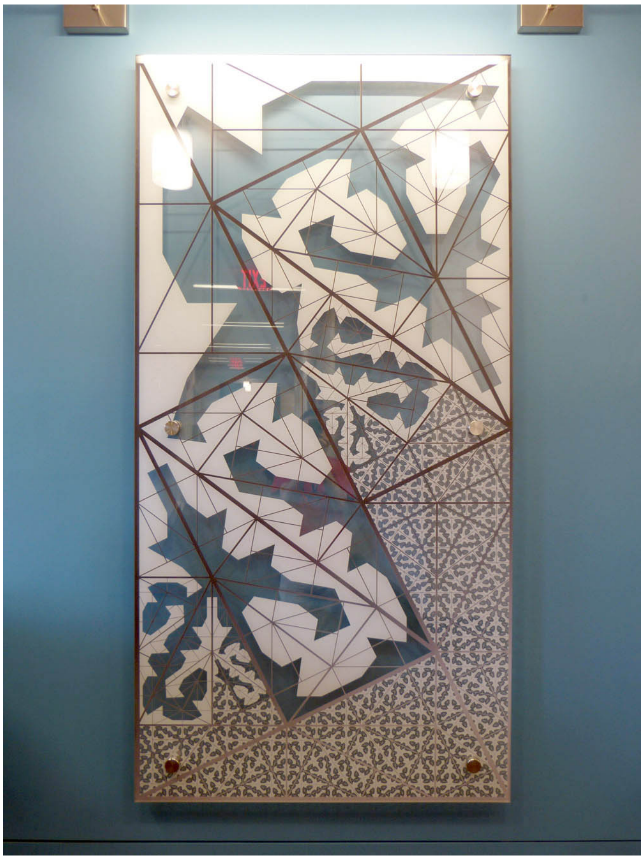

**Figure 3:** *Interest panel for the Department of Mathematics and Statistics at Colby College, designed by Henry Segerman. Photo courtesy of Henry Segerman, used with permission.*

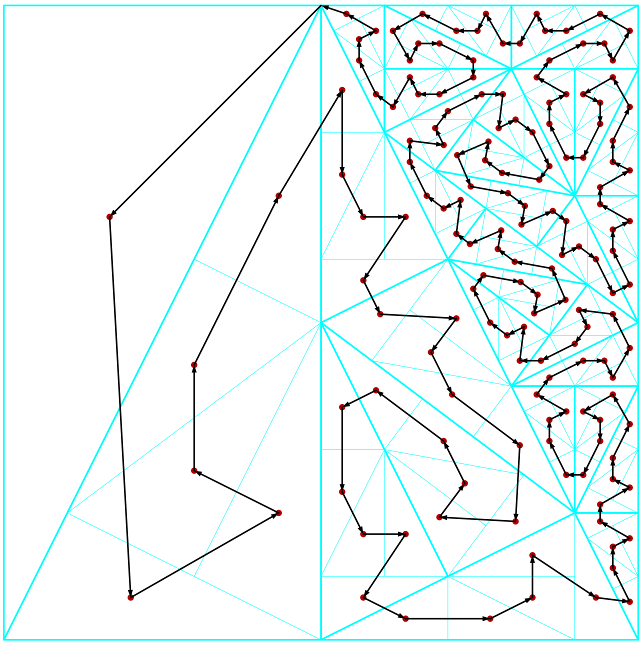

**Figure 4:** *Four iterations of pinwheel space-filling curve. In the original curve presented by Segerman, only two triangles are used (the rightmost two here), but we use four to render square images. Since the line from the last point in the rightmost triangle to the first point in the leftmost may cross the line segments in the middle triangles, we add a single auxiliary point at the top center.*

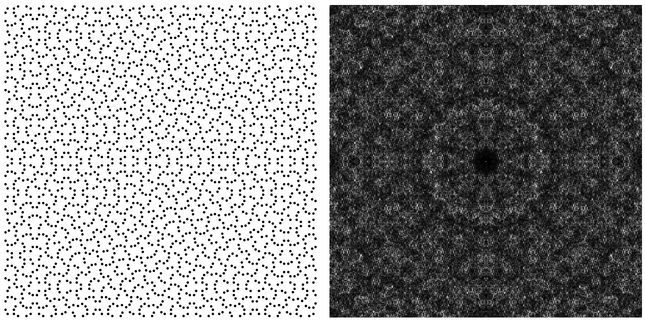


**Figure 5:** *Spatial distribution and frequency power spectrum of the centroids of a pinwheel tiling.*

to C or the reverse. The leftmost pair of tiles in Figure 4 is traversed from B to C, while the two rightmost are traversed in a reversed order.

The topmost triangular tile is retrieved from the stack for processing. A density map integrator is used to compute the total amount of "ink" (the sum of the pixels, if white is 0) in the tile, which we refer to as the mass of the tile. We use exact computation, and provide the code in the supplementary material. The mass of the tile is compared to the length of the line segment from the last generated point to the centroid of the triangle, weighted by the nominated line weight. If this segment is adequate to account for the enclosed mass then we add the centroid to the list of generated points, and proceed to process the next tile in the stack. If not, we generate five child tiles by concatenating the appropriate transformations to the parent's matrix, and insert them in the stack in the appropriate order as indicated by the *BtoC* parameter, making sure that the *BtoC* parameter is flipped for the third and fourth child. Note that the insertion order in the stack is the reverse of the processing order. The process is continued until the stack is empty. Finally, the points are connected to make the rendering.Figure

Figure 6(a) shows the subdivision and a halftoned image with the

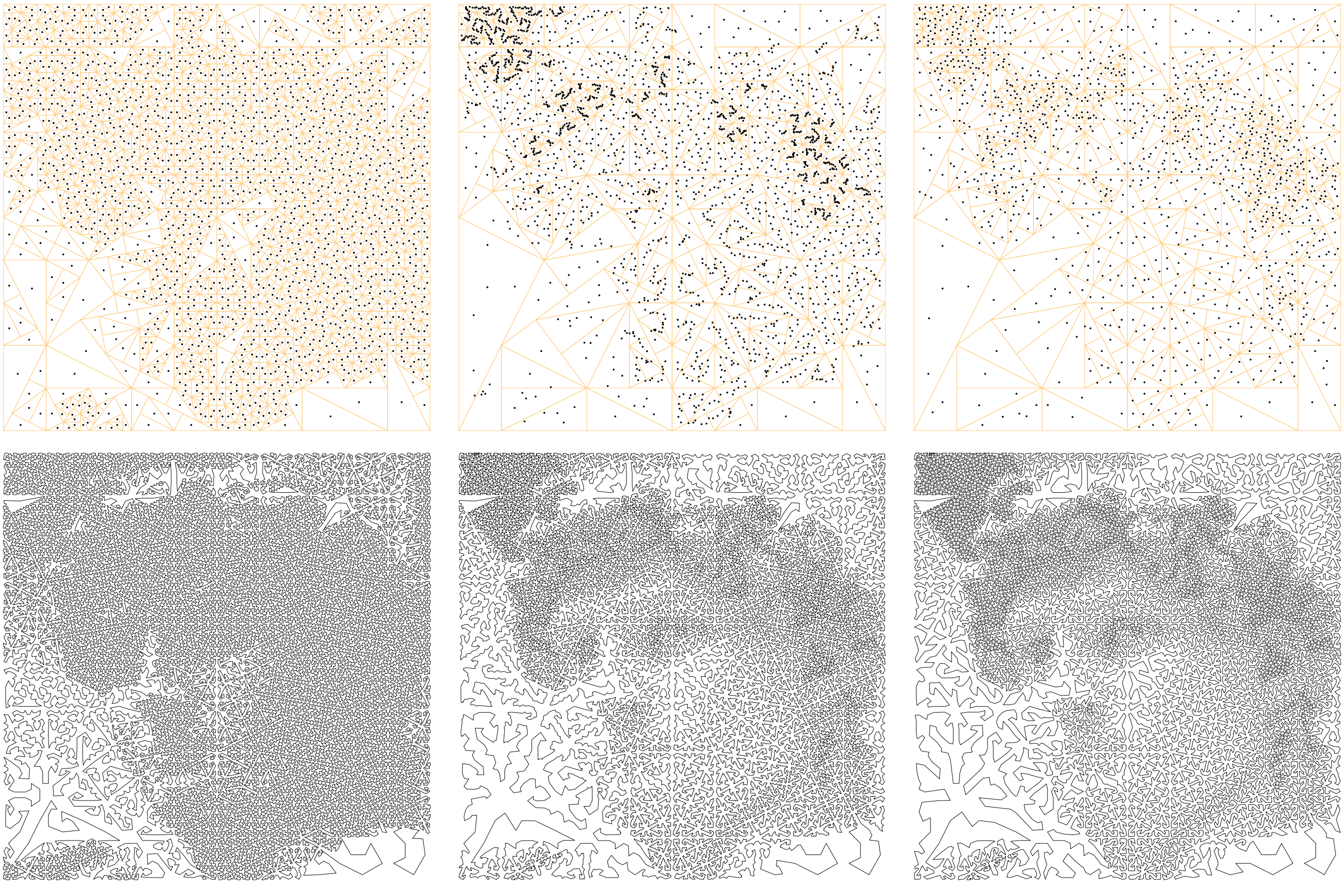


**Figure 6:** *(top) Distribution of vertices and (bottom) the resulting renderings using different levels of control. We use a subdivision depth of 5 for the top plots for the sake of clarity, and a depth of 6 in the final renderings for higher quality. (a) Subdivision only, placing a vertex at the centroid of each final-depth triangle; the distribution is smooth but banded. (b) Using five vertices at once inside each triangle to squiggle the lines; the vertices are condensed at the beginning of octaves (lighter areas of the bands in (a)), producing some noise. (c) Adding the squiggling vertices incrementally, leading to five smooth sub-bands of octaves.*

outlined algorithm. While the image is perceivable, it is not really satisfactory. Specifically, it suffers from a banding effect similar to thresholding in classic halftoning. The reason is evident: we only have a discrete set of gray tones, namely the number of subdivision levels — six in this example. We refer to these levels as *octaves*, following Ostromoukhov [Ost07].

### 3.2. Gradual Squiggling

To avoid this quantization problem we make a gradual change between the single *anchor* point in the centroid, and the five-segments line at the next level. To do that we may define five points in the segment connecting the last point to the centroid, when this segment is adequate, and gradually move each point to the corresponding centroid of the child triangle, making a longer line that can account for tones between the octaves, as illustrated in Figure 7. We use the trapezoid method to compute the appropriate interpolation such that the line length correctly matches the enclosed mass. Figure 6(b) shows the halftoned image with this improvement. Please

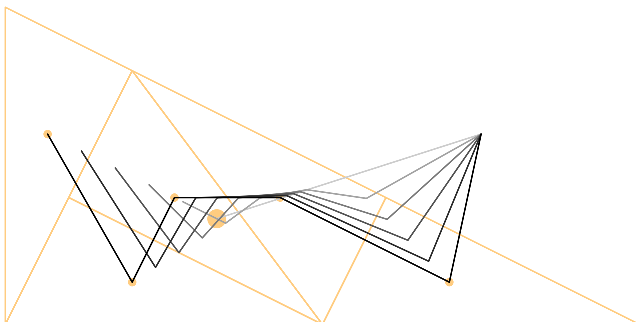

**Figure 7:** *Continuous tone can be obtained by interpolating between the single-point centroid in one octave and the five-points chain of centroids of the next octave.*

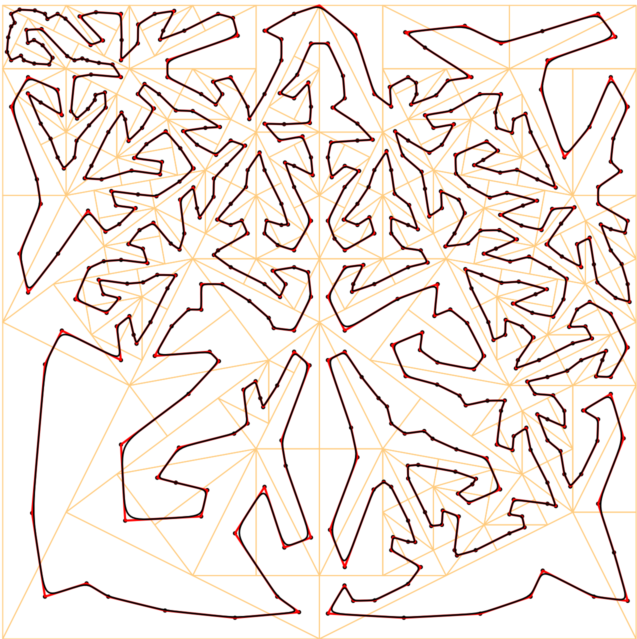

**Figure 8:** *A smooth curve is obtained by connecting the mid-points of the original line segments with Bézier curves, using the original vertices as control points.*

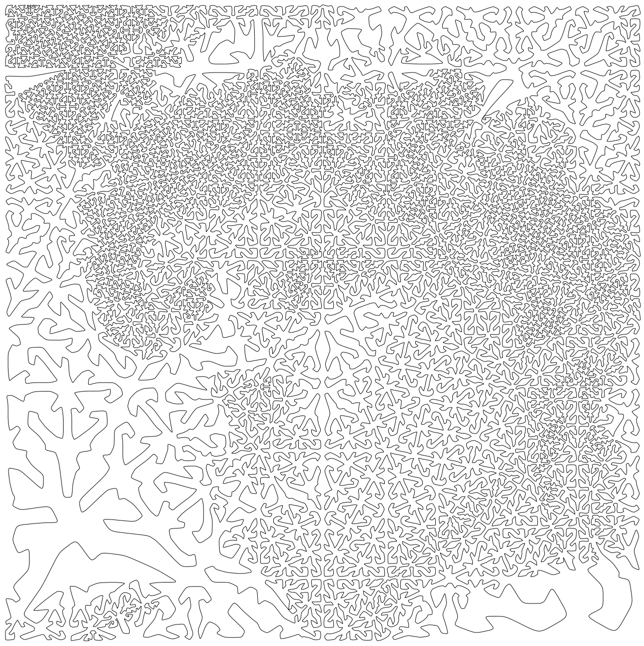

**Figure 9:** *Final rendering with smooth curves.*

look from a sufficient distance to see the image, e.g. 1.5 meters for an A4 paper.

While the mentioned squiggling approach successfully captures the tone gradients, it still suffers from a subtle problem, that in rare situations, depending on the location of the connecting points in the adjacent triangles, the line might cross itself. This violates an important and desirable feature, so it has to be avoided. To fix this problem we apply the squiggling process incrementally. We first split the single segment to the parent's centroid at the middle, and gradually move the middle point until it reaches the centroid of the entry child, either 1 when going B to C, or 5 when going C to B. To cope with more mass, we then advance the centroid of the parent towards the centroid of the exit child. Next, we split the segment between the two centroids, and gradually move the middle point towards the centroid of the middle child, 3. Finally, we split the two segments, and gradually advance them towards the centroids of the remaining child tiles, 2 and 4. This way we obtain a smooth gradual change of tones, and a smoother curve, as illustrated in Figure 6(c).

Finally, the connecting curve can be smoothed further by replacing the straight line segments with Bézier curves. For example, we may connect curved segments between midpoints of the original line segments, using the original points as control points of the curves, as illustrated in Figure 8. Figure 9 shows the final rendering with this smoothing.

### 3.3. Animation

With the gradual tone changes we introduced in the preceding subsection, our continuous-line halftoning framework becomes coherent with gradual changes in the underlying image, enabling different kinds of animations, as illustrated in Figure 1.

## 4. Discussion and Conclusion

We presented a simple yet effective approach for generating aesthetic line-based halftonings using pinwheel tiles. The method is self-contained, and is powerful enough to cope with animated pictures, producing a few frames per seconds. The performance bottleneck is in computing the mass of the triangles, and we recommend using low-resolution inputs.

While we are satisfied by the current quality, there is still a subtle discontinuity when moving from one to another octave. When the variance is high between the child tiles, it may happen that one child immediately moves to the following octave when control is passed to it. This is seen as a jump in the curve when animated. As of this writing, we were unable to find a good solution to this challenging problem, so we leave it for future research. We think this problem is challenging enough to make a subject for an exciting competition between students or programmers. The goal is to make a decision without backtracking. That being said, this problem is only related to the animated output, and does not affect still renderings.

It is worth noting that not all continuous-line drawing methods based on connecting sample points aim at halftoning. For example, the graph-based approach of Wong and Takahashi [WT11] aims at generating illustrations rather than halftonings. Adapting our presented method for this type of illustrations could be an interesting direction for future research.

**Acknowledgments:** Thanks to Henry Segerman for his comments and discussion.

## References


[AD16] AHMED A. G. M., DEUSSEN O.: Amplitude Modulated Line-Based Halftoning. In *Computer Graphics and Visual Computing (CGVC)* (2016), Turkay C., Wan T. R., (Eds.), The Eurographics Association. 2

[Ahm14] AHMED A. G. M.: Modular Line-based Halftoning via Recursive Division. In *Proceedings of the Workshop on Non-Photorealistic Animation and Rendering* (New York, NY, USA, 2014), NPAR '14, ACM, pp. 41–48. 2

[Ahm15] AHMED A. G. M.: From Stippling to Scribbling. In *Proceedings of Bridges 2015: Mathematics, Music, Art, Architecture, Culture* (Phoenix, Arizona, 2015), Kelly Delp Craig S. Kaplan D. M., Sarhangi R., (Eds.), Tessellations Publishing, pp. 267–274. Available online at `http://archive.bridgesmathart.org/2015/bridges2015-267.html`. 2

[Ahm19] AHMED A. G. M.: Sampling with Pinwheel Tiles. In *Computer Graphics and Visual Computing (CGVC)* (2019), Vidal F. P., Tam G. K. L., Roberts J. C., (Eds.), The Eurographics Association. 2

[BH04] BOSCH R., HERMAN A.: Continuous Line Drawings via the Traveling Salesman Problem. *Operations Research Letters 32*, 4 (2004), 302 – 303. 2

[Bos11] BOSCH R.: Opt Art: Special Cases. In *Proceedings of Bridges 2011: Mathematics, Music, Art, Architecture, Culture* (Phoenix, Arizona, 2011), Sarhangi R., Séquin C. H., (Eds.), Tessellations Publishing, pp. 249–256. Available online at `http://archive.bridgesmathart.org/2011/bridges2011-249.pdf`. 1

[Bro08] BROWNE C.: Truchet Curves and Surfaces. *Computers & Graphics 32*, 2 (2008), 268 – 281. 2

[dGBOD12] DE GOES F., BREEDEN K., OSTROMOUKHOV V., DESBRUN M.: Blue Noise through Optimal Transport. *ACM Trans. Graph. 31*, 6 (Nov. 2012), 171:1–171:11. 2

[IU09] INOUE K., URAHAMA K.: Chaos and Graphics: Halftoning with Minimum Spanning Trees and Its Application to Maze-like Images. *Comput. Graph. 33*, 5 (Oct. 2009), 638–647. 2

[KB05] KAPLAN C. S., BOSCH R.: TSP Art. In *Renaissance Banff: Mathematics, Music, Art, Culture* (Banff, Alberta, 2005), Sarhangi R., Moody R. V., (Eds.), Canadian Mathematical Society, pp. 301–308. Available online at `http://archive.bridgesmathart.org/2005/bridges2005-301.html`. 2

[OH95] OSTROMOUKHOV V., HERSCH R. D.: Artistic Screening. In *Proceedings of the 22Nd Annual Conference on Computer Graphics and Interactive Techniques* (New York, NY, USA, 1995), SIGGRAPH '95, ACM, pp. 219–228. 1

[Ost07] OSTROMOUKHOV V.: Sampling with Polyominoes. *ACM Trans. Graph. 26*, 3 (July 2007). 4

[PB96] PNUELI Y., BRUCKSTEIN A. M.: Gridless Halftoning: A Reincarnation of the Old Method. *Graphical Models and Image Processing 58*, 1 (1996), 38 – 64. 1, 2

[PS06] PEDERSEN H., SINGH K.: Organic Labyrinths and Mazes. In *Proceedings of the 4th International Symposium on Non-photorealistic Animation and Rendering* (New York, NY, USA, 2006), NPAR '06, ACM, pp. 79–86. 2

[Rad94] RADIN C.: The Pinwheel Tilings of the Plane. *Annals of Mathematics 139*, 3 (1994), 661–702. 2

[SB19] STOPPEL S., BRUCKNER S.: LinesLab: A Flexible Low-Cost Approach for the Generation of Physical Monochrome Art. *Computer Graphics Forum 38*, 6 (2019), 110–124. 2

[Sch83] SCHROEDER M.: The Eikonal Equation. *The Mathematical Intelligencer 5*, 1 (1983), 36–37. 2

[Seg14] SEGERMAN H.: Rep-tiles, fractal curves, 3D printing and the 4th dimension: public lecture at Colby College, 2014. 2

[Swa17] SWART D.: Morphing TSP Art. In *Proceedings of Bridges 2017: Mathematics, Art, Music, Architecture, Education, Culture* (Phoenix, Arizona, 2017), Swart D., Séquin C. H., Fenyvesi K., (Eds.), Tessellations Publishing, pp. 329–334. Available online at `http://archive.bridgesmathart.org/2017/bridges2017-329.pdf`. 2

[Uli87] ULICHNEY R.: *Digital Halftoning*. MIT Press, Cambridge, MA, USA, 1987. 1

[Uli88] ULICHNEY R.: Dithering with Blue Noise. *Proceedings of the IEEE 76*, 1 (Jan 1988), 56–79. 1

[WT11] WONG F. J., TAKAHASHI S.: A Graph-based Approach to Continuous Line Illustrations with Variable Levels of Detail. *Computer Graphics Forum 30*, 7 (2011), 1931–1939. 5