

# Iterated Averaging of Polygon Vertices

Kerry Mitchell

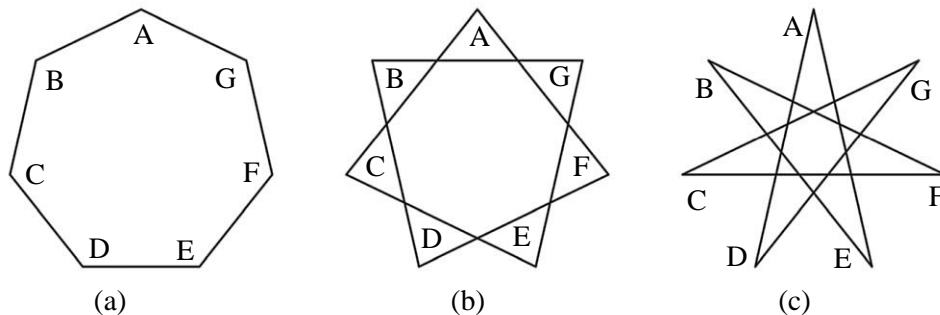
Artist, Peoria, Arizona; lkmitch@gmail.com

## Abstract

This work examines artwork generated from the problem of repeatedly averaging together the vertices of a polygon, constructing new polygons with each iteration. Rather than investigate the final state, I'm interested in the aesthetics of intermediate states. I discuss several extensions of the fundamental process, including variations on the vertices averaged, the type of average used, and the initial polygon. Considerations for creating final images and example images using these concepts are presented.

## Background

In this work, I explore artistic images of polygons that don't appear to be polygons. There are many ways to define the term "polygon," but they agree on a polygon being a closed figure on a plane whose sides are line segments. A polygon can be described as an ordered list of the  $(x,y)$ -coordinate pairs defining its vertices (corners). Using this method, the order of the points matters; Figure 1 shows three heptagons (seven sides) with the same vertices, used in different orders. In Figure 1(a), point A is the first vertex, point B the second, etc., skipping to the next vertex in line and resulting in a convex polygon. In Figure 1(b), A is first, then C, E, etc., skipping to the second next vertex in line, resulting in a non-convex polygon. In Figure 1(c), A is first, then D, G, etc., skipping to the third subsequent vertex and also leading to a non-convex polygon. In this work, polygons are defined by ordered lists of the vertices.

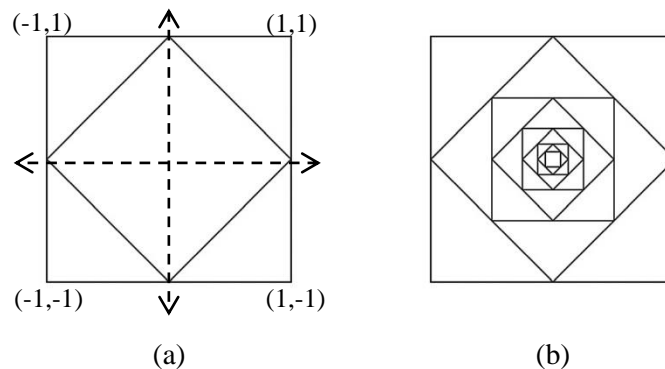


**Figure 1:** Three examples of polygons, showing the importance of vertex ordering: (a) using the vertices as listed, for a convex polygon, (b) skipping to the second next vertex for a concave polygon, (c) skipping to the third next vertex for a different convex polygon.

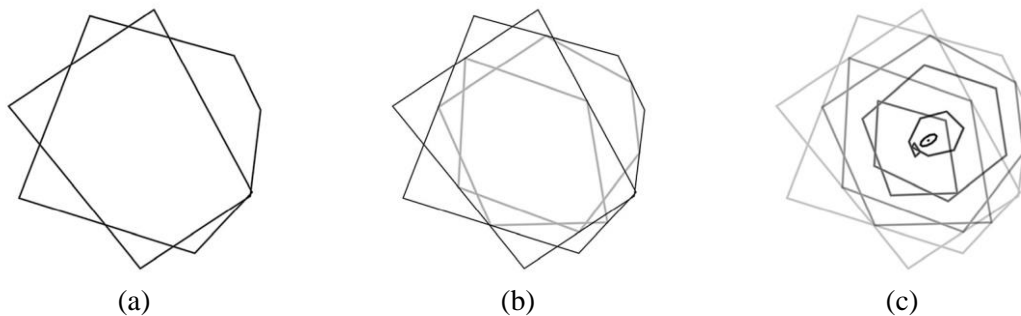
The basic process behind this work is this. Let  $P_0$  be a polygon defined by an ordered list of its vertices. Then, iteratively form new polygons  $P_k$ , where the vertices of  $P_k$  are the midpoints of the sides of polygon  $P_{k-1}$ . For example, let  $P_0$  be the square with the vertices  $\{(1,1), (-1,1), (-1,-1), (1,-1)\}$ . The midpoints of the four sides are the averages of the vertices on either end of the side, so  $P_1$  has the vertices  $\{(0,1), (-1,0), (0,-1), (1,0)\}$ , and is also a square. As shown in Figure 2, subsequent iterations of this process lead to more squares, each smaller than the last, and all centered at  $(0,0)$ .

However, if the initial polygon is random, then the subsequent polygons have different shapes and become smoother as well as smaller. Elmachoub and van Loan [3] proved that, in the limit of infinitely many iterations, the polygon's vertices lie on an ellipse centered at the centroid of the original polygon. This is illustrated in Figure 3 with a random initial polygon. Figure 3(a) shows a random decagon (10 sides).

Figure 3(b) shows the original polygon (black) and the first iteration (gray), and Figure 3(c) shows (lightest line) and the progression of iterations, from one to three, 10, 30, and 100 iterations (Even though each polygon has 10 sides, the overlapping may make some polygons appear to have fewer sides). As the number of iterations increase, the new polygons (represented by darker lines) are smaller. At 30 iterations, the polygon resembles an ellipse and the 100-iteration case is indistinguishable from a single point in the middle of the original polygon. The convergence to an ellipse is not as apparent when beginning with a regular polygon (as in Figures 2(a) and 2(b)), as the vertices of a regular polygon lie on a circle, so they have already converged to an ellipse.



**Figure 2:** Iterated averaging the vertices of a square: (a) initial square and first iteration, (b) initial square and first eight iterations.

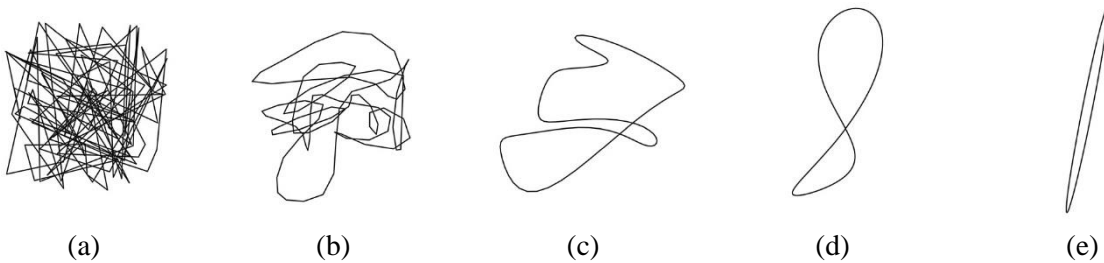


**Figure 3:** Iterated averaging the vertices of a random polygon: (a) original polygon, (b) original polygon (black) and first iteration (gray), (c) original polygon (lightest), one, three, 10, 30, and 100 iterations (darker lines correspond to more iterations).

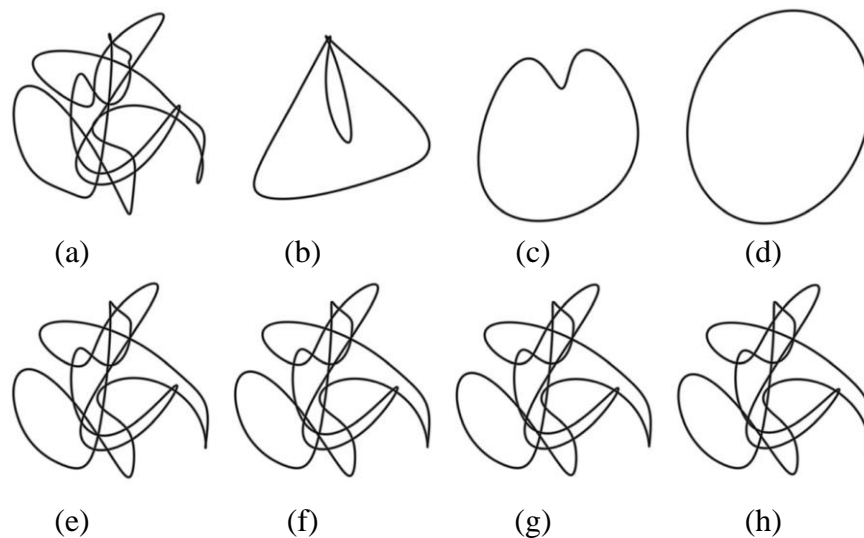
It's seen in Figures 2 and 3 that iteratively averaging the polygon vertices leads to smaller and smaller polygons, converging to a point. The mathematics behind this process have been studied extensively and is a specific example of Mean Curvature Flow [2]. Consequently, in this work, the sizes of the polygons are not of interest, only their shapes. In particular, I consider the shapes of intermediate steps between the original polygon and the limit point. Figure 4 shows an example beginning with a 100-vertex random polygon, showing the shapes of the initial polygon (Figure 4(a)), and the results after 10, 100, 1,000, and 10,000 iterations (Figures 4(b) through 4(e), respectively). In my opinion, the middle three panels of Figure 4 are more aesthetically interesting and are the subject of art arising from this method. The sections that follow discuss extensions to the basic process, which may be used for artistic effect.

### Varying the Vertices Averaged

As described above, the basic iterative process is to form new polygons whose vertices are the midpoints of the sides of the previous polygon. This is mathematically equivalent to the following: for every vertex in the old polygon, average its  $x$ - and  $y$ - coordinates with those of the next vertex and using those averages as vertex coordinates in the new polygon. There's no reason why more points can't be averaged together, say six or 10 or 20. Figure 5 shows one example, beginning with a random 900-sided polygon. (The initial polygon is essentially a jumble of random segments like Figure 4(a), but worse, and is not shown.) In Figure 5(a), two subsequent vertices are averaged at a time and the process ran for 1,800 iterations. In Figure 5(b), 5(c), and 5(d), six, 10, and 20 subsequent vertices were averaged at a time, respectively, each for 1,800 iterations. Notice how the shapes in Figures 5(b), 5(c), and 5(d) are smoother, with Figure 5(d) being very close to an ellipse. This suggests that averaging more vertices accelerates the convergence process, increasing the effective number of iterations. The effect can be compensated for by reducing the actual number of iterations. Figures 5(e) – 5(h) use the same initial random 900-sided polygon and averaged four, six, 10, and 20 vertices, respectively. However, the numbers of iterations were decreased from 450 in 5(e) to 200 in 5(f), 72 in 5(g), and 18 in 5(h), and the final shapes are almost equal. It appears that the number of iterations needed to produce approximately the same shape scales inversely with the square of the number of vertices averaged together. This conjecture was not explored further in this work.



**Figure 4:** The shapes resulting from iterated averaging the vertices of a random 100-gon: (a) original polygon, (b) 10 iterations, (c) 100 iterations, (d) 1,000 iterations, (e) 10,000 iterations.



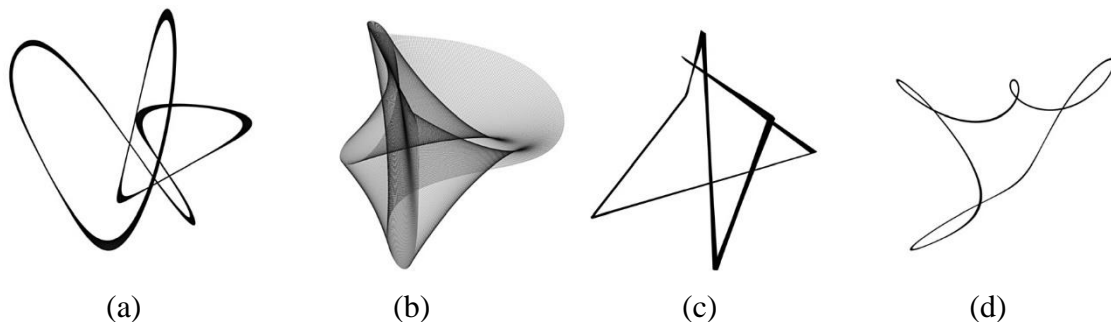
**Figure 5:** Varying the number of vertices averaged together and the number of iterations: (a) two vertices, 1,800 iterations, (b) six vertices, 1,800 iterations, (c) 10 vertices, 1,800 iterations, (d) 20 vertices, 1,800 iterations, (e) four vertices, 450 iterations, (f) six vertices, 200 iterations, (g) 10 vertices, 72 iterations, (h) 20 vertices, 18 iterations.

Another approach, which can be used independently of the number of vertices averaged together, is to change which vertices are used. For example, assume that the initial polygon has eight vertices, numbered 1 through 8. Then, the basic process would average vertices 1 and 2 together to produce vertex 1 of the new polygon, and so on, as shown in Table 1 below.

**Table 1:** Adjusting the Numbers of Vertices Skipped and Averaged Together

Basic: Skip 1, Average 2		Skip 2, Average 2		Skip 3, Average 3	
New Vertex	Old Vertices	New Vertex	Old Vertices	New Vertex	Old Vertices
1	1 & 2	1	1 & 3	1	1, 4, & 7
2	2 & 3	2	2 & 4	2	2, 5, & 8
3	3 & 4	3	3 & 5	3	3, 6, & 1
4	4 & 5	4	4 & 6	4	4, 7, & 2
5	5 & 6	5	5 & 7	5	5, 8, & 3
6	6 & 7	6	6 & 8	6	6, 1, & 4
7	7 & 8	7	7 & 1	7	7, 2, & 5
8	8 & 1	8	8 & 2	8	8, 3, & 6

Figure 6 shows four cases, each using the same initial random 1001-sided polygon and averaging together 40 vertices at a time, for 30 iterations. In Figure 6(a), 25 vertices are skipped (i.e., vertices 1, 26, 51, 76, etc. are averaged together). In Figures 6(b), 6(c), and 6(d), 30, 35, and 40 vertices are skipped, respectively. Note how the shapes in Figures 6(a) and 6(d) are relatively smooth and simple, while Figure 6(b) is decidedly more complex and Figure 6(c) is not smooth. This is apparently due to the numbers of vertices skipped in Figures 6(a) and 6(d), 25 and 40, respectively, being factors of 1,000, one less than the number of vertices. It seems that if the number of points skipped (other than one) is a factor of the number of vertices - 1 (or the number of vertices + 1), then the resulting shapes will be qualitatively different, being relatively simple and smooth.



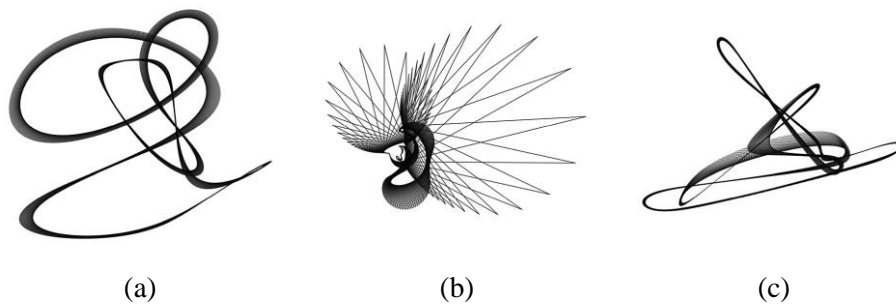
**Figure 6:** Varying the number of vertices skipped in the averaging process: (a) 25, (b) 30, (c) 35, (d) 40.

### Varying the Averaging Method

Thus far, all the cases shown have used the arithmetic mean as the averaging method. The arithmetic mean,  $A$ , is the sum of the coordinate values divided by the number of vertices used in the average. However, there are many other types of averages that can be used. The harmonic mean,  $H$ , is the reciprocal of the arithmetic mean of the reciprocal of the coordinates. That is, take each coordinate to be averaged together, and sum the reciprocals instead of the coordinates themselves. Divide this sum by number of coordinates and take the reciprocal of that result. Another is the exponential mean,  $E$ . In this case, add up the exponentials of the coordinates, divide by the number, and then take the logarithm (the inverse function of the exponential) of

the result. See [7] for other types of means. Care must be taken when using any mean other than the arithmetic mean. Coordinate values that are 0 or negative may cause computation errors, depending on the averaging method used (although that could be an interesting area of research). Also, using powers that are not positive integers, or using logarithms, may cause discontinuities in the image.

Figure 7 shows three examples. Each began with the same 1201-point random polygon, iterated 20 times, skipping 30 vertices each iteration and averaging 40 vertices at a time. In Figure 7(a), the arithmetic mean is used, the harmonic mean in 7(b) and a weighted arithmetic mean in 7(c). The spiky nature of Figure 7(b) is due to some coordinates being close to 0 and the reciprocals then becoming relatively large in magnitude. In Figure 7(c), the weight is the distance from the vertex being used in the average to the vertex whose new value is being found. For example, if the new value of vertex 1 came from averaging the old value for vertices 1, 4, and 7, then the weight for vertex 4 would be the distance cubed from the old location of vertex 1 to the old location of vertex 4.

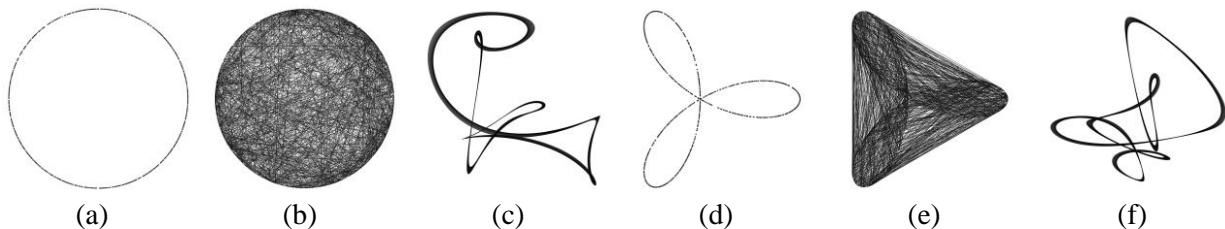


**Figure 7:** Three examples showing the use of different means: (a) arithmetic mean, (b) harmonic mean, (c) weighted arithmetic mean.

### Varying the Initial Polygon

While beginning with a random polygon is relatively easy, other sets of points can be used instead. Depending on the set used, its structure may be persistent in subsequent polygons or may be quickly diffused. For example, if the starting set is the vertices of a regular polygon (or equivalently, equally-spaced points on a circle), all subsequent shapes (using the basic process) will be regular polygons.

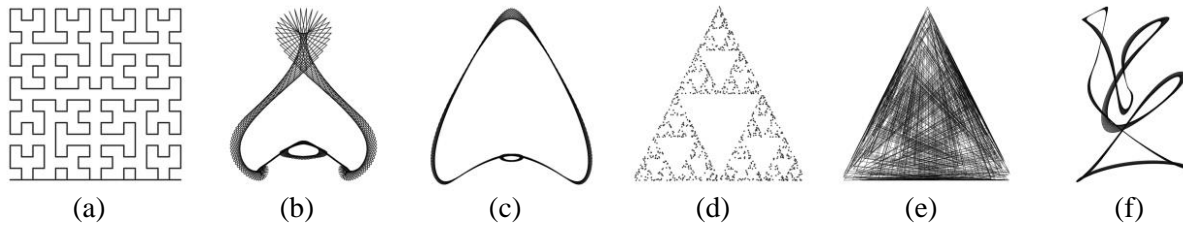
Figure 8 shows two examples using points randomly placed on well-defined curves as initial polygons. In Figures 8(a) and 8(d), 1201 points on a circle and three-lobed rose curve (respectively) serve as the initial vertices. Figures 8(b) and 8(e) show the initial polygons, and Figures 8(c) and 8(f) show the results after 20 iterations. In both cases, 30 vertices were skipped and 40 vertices were averaged using the arithmetic mean on each iteration.



**Figure 8:** Two examples using vertices randomly placed on curves: (a) initial vertices on a circle, (b) initial polygon, (c) 20 iterations, (d) initial vertices on a rose curve, (e) initial polygon, (f) 20 iterations.

Figure 9 illustrates the progression using two fractal curves, the Hilbert curve [4] and the Sierpinski triangle, created using the Chaos Game [1]. Figure 9(a) shows the initial polygon using the fourth level of the Hilbert

curve (256 vertices). Note that the last vertex (lower right corner) is connected to the first vertex (lower left corner) to close the polygon. Figures 9(b) and 9(c) show the fourth and 16th iterations, respectively, skipping 17 vertices and using the arithmetic mean to average together 14 vertices each iteration. Figure 9(d) shows 1201 initial vertices, randomly created using the Chaos Game method, and 9(e) shows the resulting initial polygon. Figure 9(f) shows the 10th iteration, skipping 30 vertices and averaging together 40 each iteration, using the arithmetic mean.



**Figure 9:** Two examples using vertices derived from fractal curves: (a) initial polygon from vertices on the Hilbert curve, (b) fourth iteration, (c) 16th iteration, (d) initial vertices on the Sierpinski triangle, (e) initial polygon, (f) 10th iteration.

## Rendering Notes and Example Images

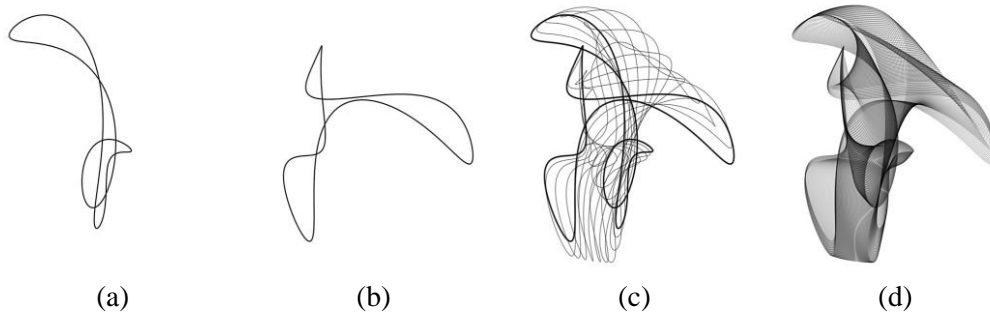
### Generating Random Numbers

The images in the paper were created using Ultra Fractal [8] and a custom coloring formula, however, the concepts can be applied in other software packages. Like many other programs, Ultra Fractal has a pseudo-random number generator, which was used to create the random vertices. Such generators usually use the linear congruential method [5], which requires an integer number as its seed. Because the integers are discrete, changing the seed slightly can lead to very different images. Figure 10 shows an example and one way to exploit that characteristic. The images in all four panels began with a 300-vertex random polygon and employed the basic method for 2,000 iterations. In Figure 10(a), the random number generator seed was 1 and in 10(b), it was 2. If you wanted an image between these two, it would be difficult to find the appropriate seed. Alternatively, the vertices of the initial random polygons can be considered two sets of points which could then be interpolated. Figure 10(c) shows 11 overlaid images on the same coordinate axes. The seed = 1 and seed = 2 shapes are drawn in bold lines. The other nine curves represent interpolations between those two, using a Bézier linear spline curve [6]. The interpolation parameter varies in from 0.0 for the seed = 1 case to 1.0 for the seed = 2 case, in steps of 0.1. The flow from one extreme to another can be imagined, but is difficult to see clearly. Figure 10(d) clarifies things by using 101 curves and varying the parameter from 0.00 to 1.00 by 0.01. Also, each curve is drawn in a light shade of gray and all are combined using a “multiply” merge mode (a gray pixel multiplied by another gray pixel becomes a darker gray pixel), so areas where multiple curves cross are darker than regions with a single curve. The Bézier curve formula works with complex numbers as well as real numbers, so the interpolation parameter can take on any complex value, greatly expanding the utility of the technique.

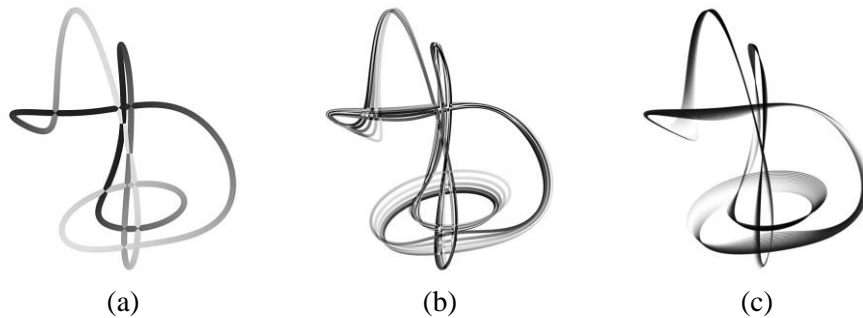
### Combining Multiple Polygon Curves into a Single Image

Figure 10 illustrates one method of combining multiple polygon curves into a single image, by simply overlaying them. Then, the manner of combining them (e.g., multiplying or averaging) is limited only by the software of choice. Also, the lines can be rendered in different ways, allowing for additional flexibility in combining them. For example, since the vertices are ordered, each segment between subsequent vertices could be rendered in a different color. Figure 11 shows an example. In Figure 11(a), a single polygon curve is drawn with the color varying with the side (segment between vertices), from dark gray to light gray and to dark again. Figure 11(b) combines five such curves. Each curve has a different combination of the number of iterations and the number of vertices averaged together. The curves are rendered as thicker lines and combined using a “difference” merge mode to highlight the changes from one to another. Figure 11(c)

shows a more subtle use of this technique. Like in the previous panel, each of the 21 curves has a slightly different combination of parameters, changing slowly from one curve to the next, so that the variations are not as apparent.

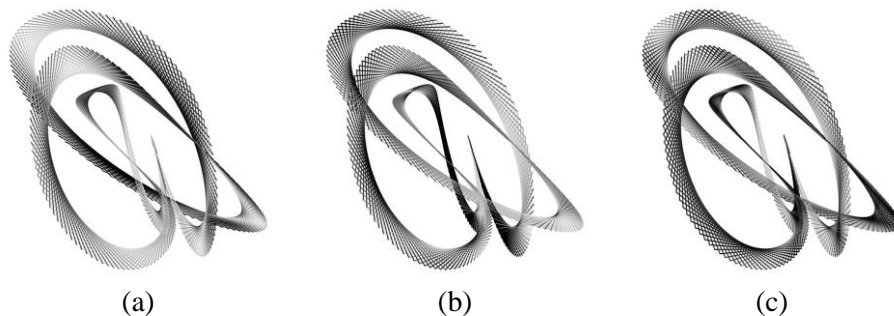


**Figure 10:** *Interpolating polygons using Bézier curves: (a) seed = 1, (b) seed = 2, (c) seed = 1 and seed = 2 (both bold) with nine interpolated curves, (d) seed = 1 and seed = 2 with 99 interpolated curves.*



**Figure 11:** *Rendering the polygon by the side number: (a) single polygon; (b) five slightly different polygons, (c) 21 different polygons.*

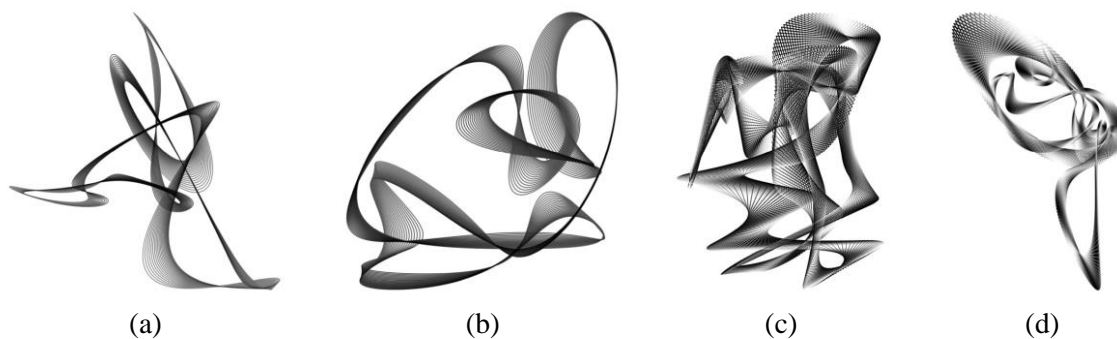
Since polygon sides are line segments, two additional aspects that can be used for coloring are the length and angle. Figure 12 shows an example for a 401-sided random polygon iterated 10 times using the arithmetic mean, skipping 20 vertices and averaging 20 vertices together at a time. In Figure 12(a), the sides are colored according to their length, and in 12(b), by their angle. Figure 12(c) combines them by taking the difference between their grayscale values.



**Figure 12:** *Rendering the polygon sides by length and angle: (a) by side length; (b) by side angle, (c) combining length and angle.*

In Figure 13, I show four example artworks that illustrate the concepts explored in this paper. Due to the constraints of printing the proceedings, the original color images have been converted to grayscale and tweaked slightly for aesthetic appeal. You may see the color works at my website, [kerrymitchell.art](http://kerrymitchell.art). Figures

13(a) and 13(b) show two companion pieces, “Transformations 1” and “Transformations 2.” They both employ 2500-side random initial polygons (with different seeds) and the arithmetic mean. Each is composed of 21 different polygons, each with a different combination of iterations and vertices averaged together (as in Figure 11(c)). The smooth nature of the polygonal curves is due to only one vertex being skipped. In contrast, the images in Figures 13(c) and 13(d) skipped many vertices each iteration (30 and 32, respectively), giving them their stellated shapes. “Dejected” in Figure 13(c) began with a 1201-sided random polygon and was iterated 12 times, averaging 20 vertices at a time. This image used a weighted arithmetic mean, where the weight was the hyperbolic tangent of the distance from the vertex being averaged to the previous value of the vertex being updated. “On Pointe” in Figure 13(d) used a random 1025-sided initial polygon, which was created using a quadratic spline between seed values. It also used a weighted arithmetic mean, but the weight here was the magnitude of the ratio of the coordinates of the vertex being updated to the coordinates of the vertex used in the average (both sets of coordinates were treated as complex numbers in forming the ratio). Eight iterations were used, averaging 32 vertices each iteration and the shape was rendered according the side number, as discussed above.



**Figure 13:** Four example artworks from the website *kerrymitchell.art*, using the concepts in this paper: (a) “Transformations 1”, (b) “Transformations 2”, (c) “Dejected”, (d) “On Pointe.”

### Summary and Conclusions

This work examined intermediate steps in the process of repeatedly averaging vertices of a polygon, before the new vertices converged to lying on an ellipse. These intermediate forms are quite pleasing to the eye and belie neither the initial polygon nor the final limiting state. Many suggestions for expanding the process were presented, including varying the points used, the manner of averaging, and the initial polygon. Examples of artworks using these ideas were presented, and it seems an area ripe for additional exploration.

### References

- [1] Barnsley, M. *Fractals Everywhere*. Morgan Kaufmann. ISBN 978-0-12-079061-6, 1993.
- [2] Colding, T.H.; Minicozzi II, W.P.; and Pedersen, E.K. “Mean Curvature Flow.” *Bulletin of the American Mathematical Society*, vol. 52, no. 2, 2015, pp. 297-333.
- [3] A. N. Elmachtoub and C. F. Van Loan. “From Random Polygon to Ellipse: An Eigenanalysis.” *SIAM Review*, vol. 52, no.1, 2010, pp. 151-170.
- [4] McKenna, D.M. *Hilbert Curves: Outside-In and Inside-Gone*. Mathemaesthetics, Inc. ISBN 978-1-7332188-0-1, 2019.
- [5] Rotenberg, A. “A New Pseudo-Random Number Generator.” *Journal of the ACM*, vol 7, no. 1, pp. 75-77.
- [6] E. W. Weisstein. “Bézier Curve.” <https://mathworld.wolfram.com/BezierCurve.html>.
- [7] E. W. Weisstein. “Mean.” <https://mathworld.wolfram.com/Mean.html>.
- [8] Ultra Fractal. <https://www.ultrafractal.com>.