SN Computer Science (2022) 3: 419
https://doi.org/10.1007/s42979-022-01343-2

ORIGINAL RESEARCH

Iterative Respacing of Polygonal Curves

Marcella Manivel[1] · Milena Silva[2] · Robert Thompson[3]

Received: 24 October 2021 / Accepted: 21 July 2022 / Published online: 4 August 2022

Abstract
A *polygonal curve* is a collection of m connected line segments specified as the linear interpolation of a list of points $\{p_0, p_1, \ldots, p_m\}$. In applications it can be useful for a polygonal curve to be *equilateral*, with equal distance between consecutive points. We present a computationally efficient method for respacing the points of a polygonal curve and show that iteration of this method converges to an equilateral polygonal curve.

Keywords Polygonal curve · Equilateral · Uniform mesh · Image processing · Shape comparison

Introduction

A *polygonal curve* (also *polygonal chain* or *polygonal path*) is a collection of m connected line segments specified by a sequence of points $\{p_0, p_1, \ldots, p_m\}$ in $\mathbb{R}^n$ called *vertices*. In computer vision, polygonal curves arise as piecewise linear representations of digital curves, and serve as effective tools for shape analysis [8]. A fundamental step in determining these representations is *polygonal curve approximation*: for a give polygonal curve C, we must find another polygonal curve $\overline{C}$—often subject to constraints or other conditions—which is sufficiently close to C. Polygonal curve approximation is often concerned with minimizing the error between C and $\overline{C}$, or minimizing the number of vertices in $\overline{C}$. Extensive literature is devoted to these objectives; most notably the classical algorithms of Ramer–Douglas–Peucker and Imai and Iri [6, 11, 12, 14].

Robert Thompson
rthompson@carleton.edu

Marcella Manivel
maniv013@umn.edu

Milena Silva
silva343@umn.edu

1 School of Mathematics, University of Minnesota, Minneapolis, USA

2 Division of Biostatistics, University of Minnesota, Minneapolis, USA

3 Department of Mathematics and Statistics, Carleton College, Northfield, USA

An *equilateral polygonal curve* is a polygonal curve for which all Euclidean distances between consecutive points, $\|p_k - p_{k-1}\|$, $k = 1, \ldots, m$, are equal. Equilateral polygonal curves are useful in the context of shape analysis for object reassembly, where they can be directly compared to determine possible congruence [3, 18]. When these equilateral curves represent the boundaries of fragmented objects, these congruences suggest how these objects may fit together. For example, this approach has been used for the automated reassembly of jigsaw puzzles and broken eggs [7, 9, 10].

The present work considers a method for polygonal approximation with equilateral (or nearly equilateral) polygonal curves. There is extensive literature on polygonal approximation by curves satisfying various constraints, e.g., [2, 4, 17], but very few that apply an equilateral constraint. One of the few existing works on this topic [15], uses a nonlinear optimization process to compute an equilateral curve that minimizes a weighted distance from vertices of the original curve. In this work, we take a different tack with a similar goal: by sampling a given polygonal curve according to a fixed arclength measurement, we create a new polygonal curve which is close to the original and has more regularly spaced vertices (illustrated in Fig. 1). As we will show, repeating this resampling process creates curves with increasing regularity in the spacing of their vertices, limiting to an equilateral curve. In contrast to most other studies of polygonal approximation, we do not attempt to minimize any kind of distance in the approximation, nor do we impose any constraints on the approximating curve. Instead our goal is to understand the effect of our repeated resampling process from theoretical and empirical viewpoints.

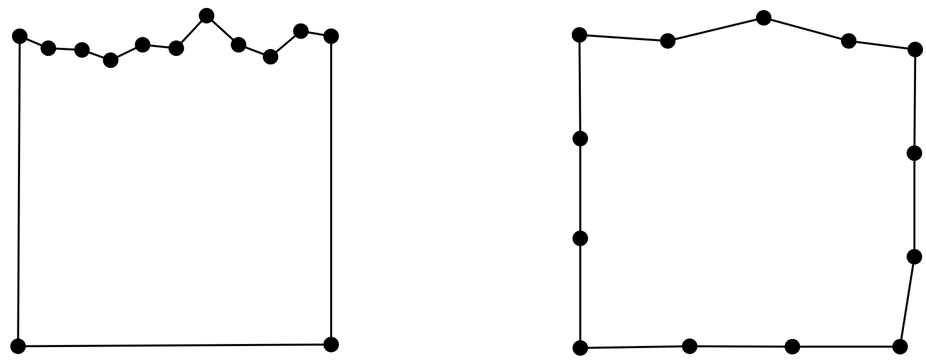


Fig. 1 Unevenly spaced polygonal curve (left) and the polygonal curve that results after one application of arclength respacing (right)

The layout of this paper is as follows. In "The Arclength Respacing Method" we introduce the arclength respacing process for a polygonal curve, outline its implementation, and establish some of its key properties. In "Iteration of the Respacing Method" we consider iteration of arclength respacing and show in Theorem 11 that the iteration tends toward a limiting polygonal curve and that the limit is equilateral. In "Examples" we examine various examples. In "Application" we illustrate the effect of arclength respacing on polygonal curves approximating a shape.

The Arclength Respacing Method

Let C be a polygonal curve with vertices $\{p_0, p_1, p_2, \dots, p_m\}$. Curves will be visualized in $\mathbb{R}^2$, but all discussion that follows applies equally to curves in $\mathbb{R}^n$. We make no restrictions on the points that define C; for example, polygonal curves may be closed (have an overlapping startpoint and endpoint) or otherwise have self-intersections or overlapping vertices. Denote by $L(C) = \sum_{k=1}^{m} \|p_k - p_{k-1}\|$ the length of C. Let $P(s)$ be the linear interpolation of the vertices of C, parameterized by arclength s, such that $0 \le s \le L(C)$. We now define the process that we will study in the following sections: the arclength respacing of a polygonal curve.

Definition 1 The *arclength respacing* of the polygonal curve C is the polygonal curve $f(C)$ with vertices $\{f(p_0), \dots, f(p_m)\}$ evenly spaced by arclength along C:

$$f(p_k) = P\left(k\frac{L(C)}{m}\right), \quad k = 0, \dots, m.$$

An example result of arclength respacing is shown in Fig. 2.

It is important to note that arclength respacing can be performed computationally without costly numerical integration or function inversion typically needed to find an arclength parameterization. The key idea is that piecewise linearity can be exploited to find $P(s)$. Let $d_0 = 0$

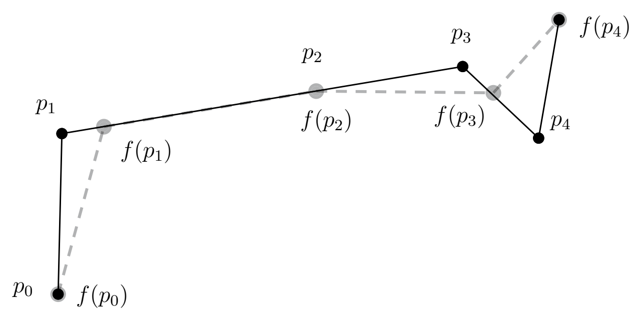


Fig. 2 Polygonal curve C and its arclength respacing $f(C)$

and $d_k = \sum_{i=1}^{k} \|p_i - p_{i-1}\|$ the arclength of C up to vertex d_k for $k = 1, \dots, m$. Then, when $d_k \neq 0$, for $0 \le t \le 1$ and $k = 1, \dots, m$, the parameterizations of segments:

$$P\big((1-t)d_{k-1} + td_k\big) = (1-t)p_{k-1} + tp_k$$

can be combined to provide an arclength parameterization $P(s)$ of C. Thus it is possible to compute $f(C)$ from C quickly using only linear interpolation. We now outline this process in more detail.

Algorithm 2 An outline of the implementation of arclength respacing. An example implementation in `Mathematica` can be found on the third author's website [19].

Input: Points $\{p_0, \dots, p_m\}$ representing the vertices of a polygonal curve C.

Output: Points $\{q_0, \dots, q_m\}$ representing the arclength respacing $f(C)$ of C.

1. Let $d_0 = 0$ and $d_k = \|p_k - p_{k-1}\| + d_{k-1}$ for $k = 1, \dots m$. d_k is the piecewise linear arclength distance from p_0 to p_k.
2. Compute the piecewise linear interpolating function $g : [0, d_m] \to [0, m]$ for the points (d_k, k), $0 \le k \le m$. This function inverts the arclength measurements, so that $g(d_k) = k$. Here we require $p_k \neq p_{k+1}$ in order for this inverse to be well defined.

3. Compute the piecewise linear interpolating function $h : [0, m] \to \mathbb{R}^2$ for the discrete curve points $p_0, \ldots, p_m$.
4. Let $\delta = d_m/m$ and define $q_k = h(g(k\delta))$ for $k = 0, 1, \ldots, m$. The points q_k are separated by an arclength distance of δ along C. These points are the vertices of $f(C)$.

Remark 3 Note that in Step 2 of Algorithm 2, the linear interpolation of the inverted arclength distances is key to avoiding inefficient arclength integral computations when performing arclength respacing. In addition, note that the distance of δ in Step 4 can be chosen to fit the application. For example, one could choose the same δ across a collection of curves to have consistent arclength spacing for curve comparison. Finally, since each step of Algorithm 2 involves a fixed time computation done at each of the m points of C, the algorithm has a run time which is $O(m)$.

Polygonal curves with different vertex sets can have the same arclength parameterized linear interpolation. To account for this ambiguity, we introduce the notion of *similar* polygonal curves.

Definition 4 A vertex p_k of a polygonal curve is called a *basic vertex* if it is not equal to the preceding vertex p_{k-1} and does not lie on the line segment between its neighboring vertices p_{k-1} and p_{k+1}. Polygonal curves P and Q are *similar*, $P \sim Q$, if they share the same sequence of basic vertices. Note that the initial vertex p_0 is always basic. Two similar polygonal curves are shown in Fig. 3.

Lemma 5 *Similar polygonal curves have the same arclength parameterized linear interpolation.*

Proof If p_k is a non-basic vertex, either $p_{k-1} = p_k$, or p_k lies in the image of the line segment connecting p_{k-1} and p_{k+1}. In either case, removing p_k from the vertex set of C will not affect the arclength parameterized linear interpolation (we either remove a segment of length zero or replace two segments with one segment of combined length). Thus the arclength parameterized linear interpolation is determined only by the basic vertices, and similar curves have the same basic vertices. □

We now investigate how the process of arclength respacing changes a polygonal curve. In particular, we find in Proposition 8 that similarity characterizes particular ways that the length and spacing of a polygonal curve change under arclength respacing. We first prove a simple lemma about a more general respacing process we call *oriented resampling*, of which arclength respacing is a special case.

Definition 6 Let C be a polygonal curve and $P(s)$, $0 \le s \le L(C)$, the associated arclength parameterized linear interpolation of the vertices of C. Suppose we sample $m + 1$ values of arclength $0 = s_0 \le s_1 \le \cdots \le s_{m-1} \le s_m = L(C)$ and define a corresponding sample of points $q_0, \ldots, q_m$ along C such that $q_k = P(s_k)$, for $k = 0, \ldots, m$. The polygonal curve D defined by the vertex set $q_0, \ldots, q_m$ is called an *oriented resampling* of P.

Lemma 7 *If D is an oriented resampling of C, then $L(D) \le L(C)$. In particular, $L(f(C)) \le L(C)$, so arclength respacing cannot increase the length of a polygonal curve.*

Proof We use the notation of Definition 6. First observe that $\|q_k - q_{k-1}\| \le s_k - s_{k-1}$, since $s_k - s_{k-1}$ is the arclength distance between points q_{k-1} and q_k measured along C. Thus

$$
\begin{aligned}
L(D) &= \sum_{k=1}^{m} \|q_k - q_{k-1}\| \\
&\le \sum_{k=1}^{m} s_k - s_{k-1} \\
&= s_m - s_0 = L(C).
\end{aligned}
$$

□

Proposition 8 *Let C be a polygonal curve and $f(C)$ the arclength respacing of C. The following properties are equivalent*:

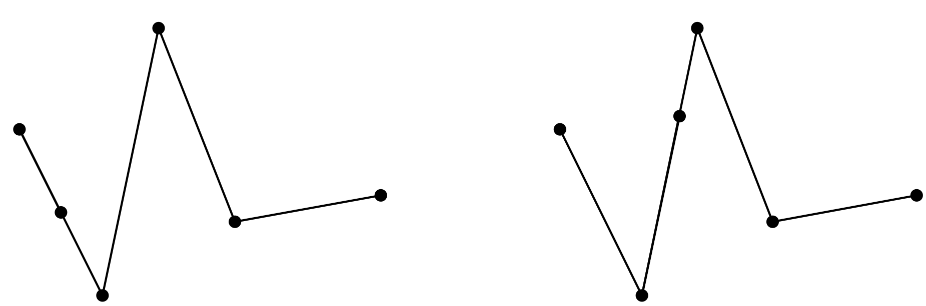


Fig. 3 Two similar polygonal curves with different vertices

(1) $L(C) = L(f(C))$,
(2) $C \sim f(C)$,
(3) $f(C)$ *is equilateral.*

Proof We first show (1) $\iff$ (2). If $C \sim f(C)$ then $L(f(C)) = L(C)$ by Lemma 5. Suppose then that $f(C) \nsim C$. There must then be a vertex p_k of C which is not equal to any vertex of $f(C)$, nor lies on a line segment connecting two consecutive vertices of $f(C)$. Suppose that q_n and q_{n+1} are the vertices of $f(C)$ immediately preceding and following p_k along C. Let p_l be the vertex of C immediately preceding q_n along C and p_r the vertex immediately following q_{n+1} along C. Let

$$\overline{C} = \{p_0, \dots, p_l, q_n, q_{n+1}, p_r, \dots, p_m\}$$

be the polygonal curve obtained from C by adding the vertices q_n, q_{n+1} to C and deleting all vertices $p_{l+1}, \dots, p_k, \dots, p_{r-1}$ between q_n and q_{n+1} along C. Because these deleted vertices do not all lie along the line segment connecting q_n and q_{n+1} (in particular, p_k does not), we have the strict inequality: $L(\overline{C}) < L(C)$. Since the arclength respacing $f(C)$ is also an oriented resampling of $\overline{C}$, $L(f(C)) \le L(\overline{C})$ by Lemma 7. Combining these two inequalities gives $L(f(C)) < L(C)$.

We next show (2) $\iff$ (3). Suppose first that $C \sim f(C)$. Then, by Lemma 5, C and $f(C)$ have the same arclength parameterized linear interpolation. Thus a sampling of points evenly spaced along C by arclength will also be evenly spaced by arclength along $f(C)$, so $f(C)$ will be equilateral.

Assume now that $f(C)$ is equilateral. Because there are $m+1$ points in $f(C)$ and m line segments in C, there must be a pair of consecutive vertices q_k, q_{k+1} of $f(C)$ which lie on the same line segment of C. Thus $\|q_{k+1} - q_k\| = L(C)/m$. Because $f(C)$ is equilateral, we also have $\|q_{k+1} - q_k\| = L(f(C))/m$. Thus $L(C) = L(f(C))$, and so $C \sim f(C)$, since we have already established that (1) $\implies$ (2). □

We now examine what happens when the arclength respacing process is repeated. First observe that, although the vertices of $f(C)$ are equally spaced by arclength along C, they are not necessarily equally spaced by arclength along $f(C)$. Thus $f(C)$ is not necessarily an equilateral polygonal curve. Nevertheless, it appears that vertices of $f(C)$ are more evenly spaced than those of C (as seen clearly in Fig. 1). Thus, in an effort to obtain an equilateral curve, we can iterate the arclength respacing process, hoping to obtain polygonal curves that are increasingly close to being equilateral. The result of iterated arclength respacing is considered in the next section.

Iteration of the Respacing Method

Let C be a polygonal curve and denote by C^n the n^{th} iteration of the arclength respacing: $C^n = f^n(C)$. Lemma 7 yields an immediate observation about this iteration: the sequence of lengths of the iterated curves must converge.

Lemma 9 *$L(C^n)$ converges as $n \to \infty$.*

Proof From Lemma 7, the sequence $\{L(C^n)\}$ is nonincreasing. This sequence is also bounded below by the distance $\|p_m - p_0\|$ between the endpoints of C. (Note that this could be 0 for a closed polygonal curve.) Thus $\{L(C^n)\}$ is bounded and monotonic and must converge. □

As demonstrated in Proposition 8, arclength respacing yields an equilateral curve only in special cases. However, arclength respacing generally produces a curve that appears closer to being equilateral (as shown in Fig. 2, for example). More precisely, as the respacing is repeated, the iterates will converge to an equilateral curve. We introduce an equivalent definition of equilateral that will be useful in proving this.

Lemma 10 *$C = \{p_0, \dots, p_m\}$ is equilateral if and only if, for $k = 1, \dots, m$:*

$$\frac{kL(C)}{m} = \sum_{i=1}^{k} \|p_i - p_{i-1}\|.$$

Theorem 11 *For any polygonal curve C, $\lim_{n\to\infty} f^n(C)$ exists and is an equilateral curve.*

Proof We will argue that, for each p_k, the sequence $\{f^n(p_k)\}_{n=0}^{\infty}$ converges, and thus that the limiting polygonal curve is given by $C^* = \{p_0^*, \dots, p_m^*\}$, where $p_k^* = \lim_{n\to\infty} f^n(p_k)$. For convenience, we will use the notation $p_k^n = f^n(p_k)$ in what follows.

Let $\epsilon > 0$. By Lemma 9 there is an N such that for all $n \ge N$:

$$L(C^n) - \epsilon < L(C^{n+1}) \le L(C^n). \tag{1}$$

Consider the distance $\|p_k^{n+1} - p_{k-1}^{n+1}\|$. This distance must satisfy the inequality:

$$\frac{L(C^n)}{m} - \epsilon < \|p_k^{n+1} - p_{k-1}^{n+1}\| \le \frac{L(C^n)}{m} \tag{2}$$

for $n \ge N$ and $k = 1, \dots, m$. The right side of this inequality is by construction, since the points p_k^{n+1} are chosen to be evenly spaced by arclength distance $\frac{L(C^n)}{m}$ along C^n. For the left side, assume that there is some k^* such that $\|p_{k^*}^{n+1} - p_{k^*-1}^{n+1}\| \le \frac{L(C^n)}{m} - \epsilon$. Then

$$
\begin{aligned}
L(C^{n+1}) &= \sum_{k=1}^{m} \|p_k^{n+1} - p_{k-1}^{n+1}\| \\
&= \|p_{k^*}^{n+1} - p_{k^*-1}^{n+1}\| + \sum_{k \neq k^*} \|p_k^{n+1} - p_{k-1}^{n}\| \\
&\leq \frac{L(C^n)}{m} - \epsilon + (m-1)\frac{L(C^n)}{m} \\
&< L(C^{n+1}),
\end{aligned}
$$

where the last two inequalities follow from (1). This is a contradiction, and (2) follows.

Next, for $n \geq N$ and $k = 1, \ldots, m$, consider the distance between points on consecutive iterates, $\|p_k^{n+2} - p_k^{n+1}\|$. By construction, the point p_k^{n+2} is placed on C^{n+1} an arclength distance of $\frac{kL(C^{n+1})}{m}$ from p_0^{n+1} along C^{n+1}. By definition, the point p_k^{n+1} lies an arclength distance of $\sum_{i=1}^{k} \|p_i^{n+1} - p_{i-1}^{n+1}\|$ from p_0^{n+1} along C^{n+1}. Thus, the Euclidean distance $\|p_k^{n+2} - p_k^{n+1}\|$ is bounded by the difference of these arclength distances along C^{n+1}:

$$
\|p_k^{n+2} - p_k^{n+1}\| \leq \left| \frac{kL(C^{n+1})}{m} - \sum_{i=1}^{k} \|p_i^{n+1} - p_{i-1}^{n+1}\| \right|. \tag{3}
$$

With some rearrangement we find

$$
\begin{aligned}
&\left| \frac{kL(C^{n+1})}{m} - \sum_{i=1}^{k} \|p_i^{n+1} - p_{i-1}^{n+1}\| \right| \\
&\quad = \left| \sum_{i=1}^{k} \frac{L(C^{n+1})}{m} - \|p_i^{n+1} - p_{i-1}^{n+1}\| \right| \\
&\quad \leq \sum_{i=1}^{k} \left| \frac{L(C^{n+1})}{m} - \|p_i^{n+1} - p_{i-1}^{n+1}\| \right|.
\end{aligned} \tag{4}
$$

Now, by (1)

$$
\frac{L(C^n)}{m} - \epsilon < \frac{L(C^{n+1})}{m} \leq \frac{L(C^n)}{m},
$$

which, combined with (2), yields

$$
\left| \frac{L(C^{n+1})}{m} - \|p_i^{n+1} - p_{i-1}^{n+1}\| \right| < \epsilon
$$

for $i = 1, \ldots, k$. Thus

$$
\begin{aligned}
\|p_k^{n+2} - p_k^{n+1}\| &\leq \sum_{i=1}^{k} \left| \frac{L(C^{n+1})}{m} - \|p_i^{n+1} - p_{i-1}^{n+1}\| \right| \\
&< k\epsilon.
\end{aligned} \tag{5}
$$

Since m is fixed in the respacing process, this provides a uniform bound $\|p_k^{n+2} - p_k^{n+1}\| < m\epsilon$ for $k = 1, \ldots, m$ and $n \geq N$, and thus $\lim_{n\to\infty} p_k^n$ converges for $k = 0, \ldots, m$. (The $k = 0$ case is immediate, since the initial point is always fixed.) As before, call the limit p_k^* and the limiting curve $C^* = \{p_0^*, \ldots, p_m^*\}$.

Finally, we show that C^* is equilateral. From (4) and (5) we see that, for $k = 1, \ldots, m$,

$$
\lim_{n\to\infty} \left| \frac{kL(C^{n+1})}{m} - \sum_{i=1}^{k} \|p_i^{n+1} - p_{i-1}^{n+1}\| \right| = 0.
$$

Passing the limit, we have

$$
\left| \frac{kL(C^*)}{m} - \sum_{i=1}^{k} \|p_i^* - p_{i-1}^*\| \right| = 0,
$$

and thus by Lemma 10, C^* is equilateral. □

Examples

Any given polygonal curve will limit to an equilateral curve; however, it appears to be difficult in general to determine the limiting equilateral curve. In this section, we look at some examples, where we know the limiting curve or can determine information about it. We will continue to use the notation $p_k^n = f^n(p_k)$ and $p_k^* = \lim_{n\to\infty} p_k^n$ from the proof of Theorem 11.

Example 12 There are polygonal curves which become equilateral precisely at iteration n. Consider three colinear vertices p_0, p_1, p_2, with p_2 on the line segment connecting p_0 and p_1. Let $d = \|p_2 - p_0\|$. After each iteration, p_0 and p_2 remain fixed, and p_1 moves a distance $d/2$ closer to p_0. If $L(C) = n\,d$, then C will become equilateral at precisely iteration n. This example is illustrated in Fig. 4.

It is not necessary to have colinear vertices to produce an example which becomes equilateral after finitely many iterations. Figure 5 shows a curve which becomes equilateral after exactly two iterations. It is an open question if it is possible to generalize an approach like the one shown in Fig. 5 to work for an arbitrary iteration n.

Example 13 We next consider triangles, viewed as closed polygonal curves with four vertices $\{p_0, p_1, p_2, p_3 = p_0\}$. By Theorem 11, the only possible limiting polygonal curves are a single point, or an equilateral triangle. We consider starting curves which will realize either of these possibilities.

Consider an isosceles triangle C with angle $\theta > \frac{\pi}{3}$ at p_0, as shown in Fig. 6. The points p_1 and p_2 will be mapped symmetrically under iteration to points on the side opposite to

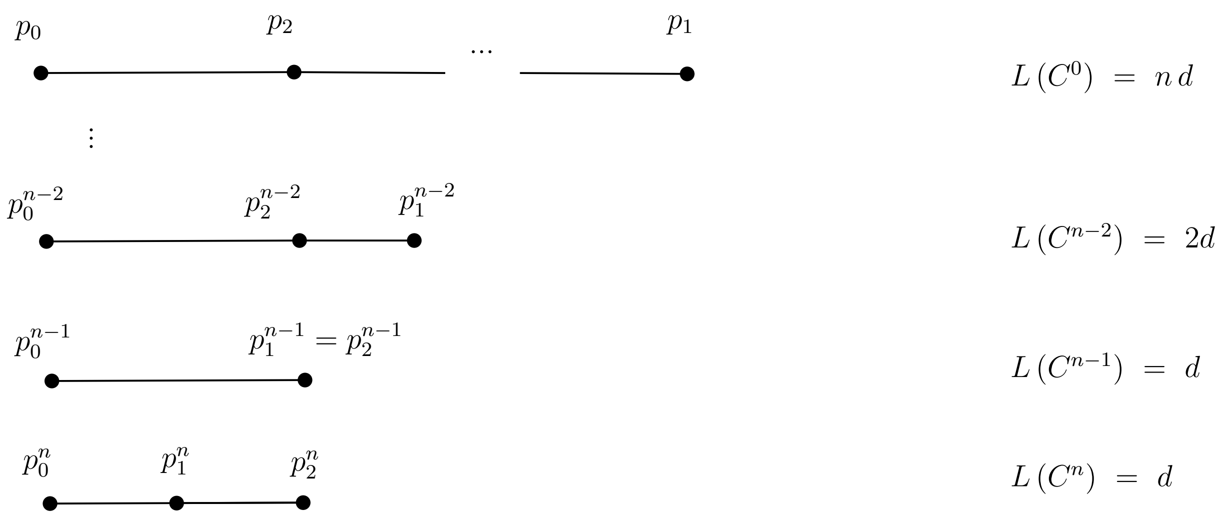


Fig. 4 Polygonal curve that becomes equilateral after n iterations

Fig. 5 Polygonal curve that becomes equilateral after two iterations

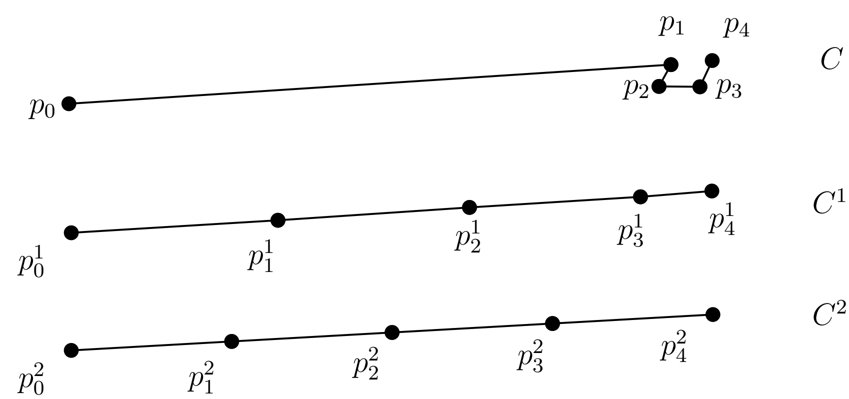
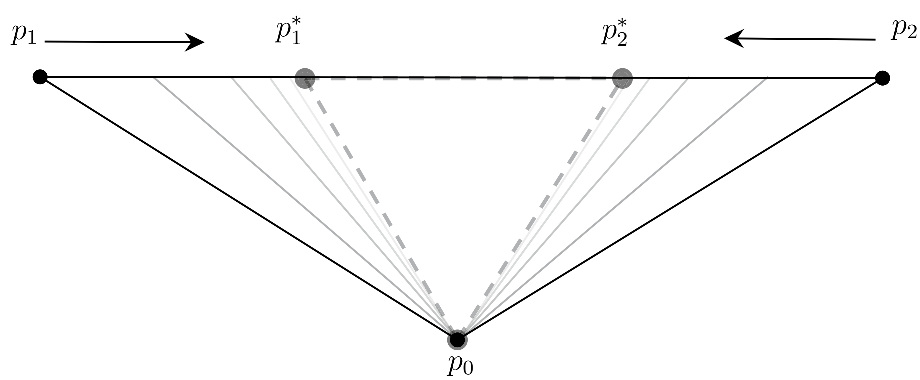


Fig. 6 Arclength respacing iteration for an isosceles triangle with angle $\theta > \frac{\pi}{3}$ at p_0 converges to an equilateral triangle



p_0. The angle θ^n at p_0^n will approach $\pi/3$ and C will approach an equilateral triangle.

Next consider a triangle C with an angle $\theta < \frac{\pi}{3}$ at the starting vertex p_0. By construction, this angle does not increase under iteration, so $\theta^{n+1} \leq \theta^n$, where θ^n is the angle at the vertex p_0^n of C^n. Thus C^n cannot approach an equilateral triangle, and must converge to a point. A special case of this is the iteration of an isosceles triangle with angle $\theta < \frac{\pi}{3}$ at p_0, which will produce a sequence of similar triangles shrinking to a point, as shown in Fig. 7.

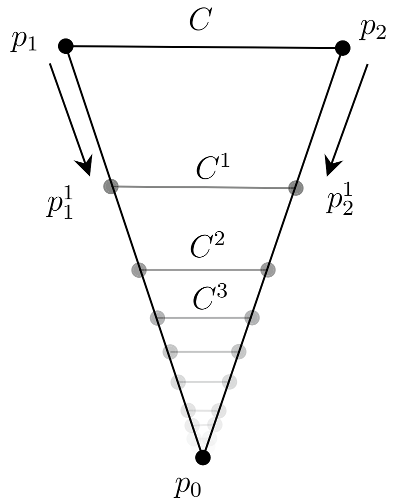


Fig. 7 Arclength respacing iteration for a triangle with angle $\theta < \frac{\pi}{3}$ at p_0 converges to a point

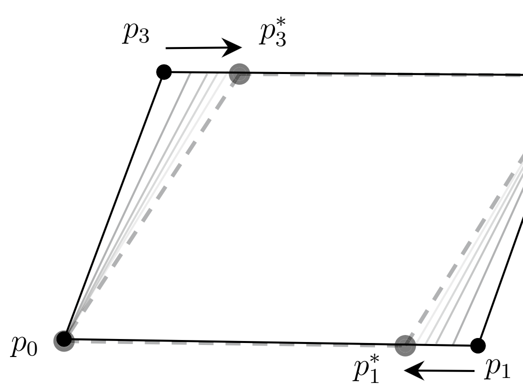


Fig. 8 Parallelogram will limit towards a rhombus under iteration

Example 14 We consider quadrilaterals, viewed as closed polygonal curves with five vertices $\{p_0, p_1, p_2, p_3, p_4 = p_0\}$. By Theorem 11, a quadrilateral must tend toward an equilateral polygonal curve, in this case a rhombus, degenerate rhombus, or a single point. If the polygonal curve $\{p_0, p_1, p_2\}$ has the same length as the polygonal curve $\{p_2, p_3, p_4\}$, the vertex p_2 will remain fixed under iteration. The limiting curve will be a rhombus. A special case of this is the iteration of a parallelogram, shown in Fig. 8.

Application

In this section, we empirically explore the practical effect arclength respacing has on polygonal curves approximating a given shape. We compute several measurements at the n^{th} iteration of the arclength respacing to quantify this effect. Iterations should tend toward an equilateral curve, so we find the standard deviation σ^n of the collection of distances $d_k^n = \|p_k^n - p_{k-1}^n\|$, $k = 1, \ldots, m$, the maximum interpoint distance $\max^n = \max_{1 \le k \le n} d_k^n$, and the minimum interpoint distance $\min^n = \min_{1 \le k \le n} d_k^n$, as different ways to measure this effect. It is also important to understand how successive iterations retain fidelity to the original polygonal curve. Since the ϵ-tolerance zone—the region consisting of the union of all radius ϵ disks centered on the polygonal curve—is a standard metric in polygonal curve approximation [5, 11, 13], we compute ϵ^n, the smallest ϵ for which the nth iterate lies within the ϵ-tolerance zone of the original polygonal curve, as measure for the accuracy of our polygonal approximation.

Shown in Fig. 9 is a synthetic "noisy cat" curve after 0, 1, and 5 applications of respacing. The noisy cat curve consists of 65 points, with x, y values in the range $[-1, 1]$. The points are visually equilateral after only a few iterations. The ears (and other smaller protrusions) become rounded, causing ϵ^n to increase sharply, but this effect quickly diminishes. Note that the jagged point on the right side of the figure does not smooth out; this is an artifact of the choice of starting point for the respacing. As seen in Fig. 10 the standard deviation σ^n appears to decrease exponentially with a factor of about 0.54. After approximately 15 iterations, the standard deviation is essentially 0 and the maximum and minimum are equal up to first five decimal places.

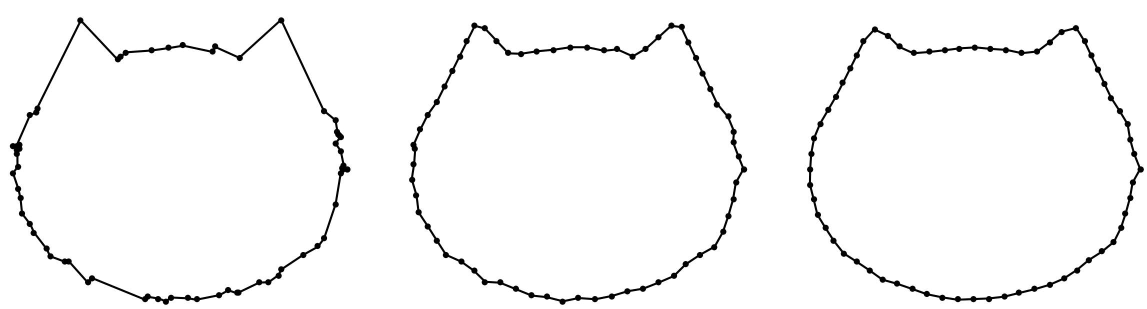


Fig. 9 "Noisy cat" polygonal curve after 0, 1, and 5 iterations of arclength respacing

We next apply arclength respacing to the outline of a jigsaw puzzle piece obtained from image segmentation, shown in Fig. 11. This jigsaw curve consists of 400 points, with x, y values in $[-6, 6]$. The vertices have been normalized so that mean interpoint distances are comparable to the "noisy cat" example, with $\delta \approx 0.1$. One iteration of arclength respacing produces a curve which appears very close to equilateral. In Fig. 12 are the basic statistics for this arclength respacing iteration. Once again, the standard deviation σ^n appears to decrease exponentially, and the standard deviation is 0 and the maximum and minimum are equal up to first five decimal places after approximately 15 iterations. The size ϵ^n of the ϵ-tolerance zone settles quickly at a value that is very small relative to the overall size of the original polygonal curve, suggesting that this iteration provides an equilateral approximation with good fidelity.

Conclusions

This paper introduces an arclength respacing method for polygonal curves and establishes general facts about the behavior of polygonal curves under iteration of this respacing. There are several interesting directions for further investigation.

The impetus for this research was the application of arclength respacing to problems in computer vision. For this application it would be very useful to better understand the rate of convergence to an equilateral polygon and the smoothing effect, the latter being especially apparent in Fig. 9. One might also consider replacing the underlying polygonal curve with a higher order interpolating function, as is done in [9], to affect rates of convergence and smoothing.

n	0	1	2	3	5	10	15
σ^n	0.127073	0.01431	0.00342	0.00117	0.00039	0.00002	0.00000
σ^n/σ^{n-1}	-	0.112646	0.23937	0.34217	0.66789	0.53647	0.53571
$\max^n$	0.65736	0.11137	0.10436	0.10274	0.10217	0.10199	0.10198
$\min^n$	0.010769	0.02796	0.08312	0.09420	0.09923	0.10184	0.10197
ϵ^n	0	0.030064	0.033578	0.033920	0.036121	0.038718	0.038843

Fig. 10 Measuring the respacing of the "noisy cat" polygonal curve

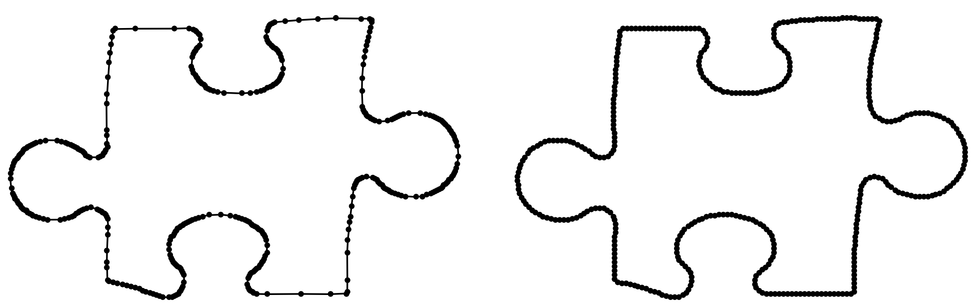


Fig. 11 Outline of a puzzle piece after 0 and 1 iteration

n	0	1	2	3	5	10	15
σ^n	0.11327	0.00123	.00038	0.00011	0.00001	0.00000	0.00000
σ^n/σ^{n-1}	-	0.01084	0.31176	0.29806	0.29953	0.40801	0.58714
$\max^n$	0.96667	0.10400	0.10369	0.10362	0.10358	0.10358	0.10358
$\min^n$	0.03018	0.08805	0.09861	0.10218	0.10341	0.10357	0.10357
ϵ^n	0	0.025417	0.032453	0.034892	0.035567	0.035705	0.035706

Fig. 12 Measuring the respacing of a puzzle piece polygonal curve

Simple, concrete examples of limiting polygonal curves are provided in "Examples", but we found it generally difficult to determine limiting curves exactly. Further investigation may discover ways to obtain information about the limiting curves, e.g., bounds for final locations of vertices or knowledge of whether the vertices will converge to a single point. In addition, one could study stability—ways in which perturbations to the initial vertex configuration affect the limiting curve.

Finally, the arclength respacing iteration bears a resemblance to the *pentagram map* [16], an operation defined on convex polygons with many interesting properties, including discrete integrability and a continuum limit corresponding to the Boussinesq equation [1, 20]. It would be worthwhile to investigate this resemblance. A first step could be understanding the continuum limit of arclength respacing; does the infinitesimal motion of the vertices produce a non-trivial curve flow? This curve flow would likely be non-local, since the respacing process is non-local.

Acknowledgements We would like to thank the Carleton College Towsley Endowment, the Louis Stokes Alliances for Minority Participation, the Carleton Summer Science Fellowship with funding from Carleton College S-STEM (NSF 0850318 and 156018), and North Star STEM Alliance, an LSAMP alliance (NSF 1201983), for providing the funding that made this research possible. We would also like to thank Peter Olver for helpful comments and advice.

Data Availability The data used to generate Figs. 9 and 10 is available from the corresponding author's website, [19]. All other data and code is available upon request.

Declaration

Conflict of interest The authors have no conflicts of interest to declare.

References

1. Beffa GM. On generalizations of the pentagram map: discretizations of AGD flows. J Nonlinear Sci. 2013;23(2):303–34.
2. de Berg M, van Kreveld M, Schirra S. Topologically correct subdivision simplification using the bandwidth criterion. Cartogr Geogr Inf Syst. 1998;25(4):243–57.
3. Bunke H, Bühler U. Applications of approximate string matching to 2D shape recognition. Pattern Recognit. 1993;26(12):1797–812.
4. Chen D, et al. Polygonal path simplification with angle constraints. Comput Geom. 2005;32(3):173–87.
5. Daescu O, Mi N. Polygonal chain approximation: a query based approach. Comput Geom. 2005;30(1):41–58.
6. Douglas DH, Peucker TK. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. Cartogr Int J Geogr Inf Geovis. 1973;10(2):112–22.
7. Grim A, O'Connor T, Olver PJ, Shakiban C, Slechta R, Thompson R. Automatic reassembly of three-dimensional jigsaw puzzles. Int J Image Graph. 2016;16(02):1650009.
8. Gomes J, Velho L, Costa Sousa M. Computer graphics: theory and practice. Boca Raton: CRC Press; 2012.
9. Hoff DJ, Olver PJ. Automatic solution of jigsaw puzzles. J Math Imaging Vis. 2014;49(1):234–50.
10. Illig P, Thompson R, Yu Q. Application of integral invariants to apictorial jigsaw puzzle assembly (2021). arXiv:2109.06922.
11. Imai H, Iri M. Computational-geometric methods for polygonal approximations of a curve. Comput Vis Graph Image Process. 1986;36(1):31–41.
12. Imai H, Iri M. An optimal algorithm for approximating a piecewise linear function. J Inf Process. 1986;9(3):159–62.
13. Melkman A, O'Rourke J. On polygonal chain approximation. Mach Intell Pattern Recognit 87–95 (1988). APA
14. Ramer U. An iterative procedure for the polygonal approximation of plane curves. Comput Graph Image Process. 1972;1(3):244–56.
15. Rannou F, Gregor J. Equilateral polygon approximation of closed contours. Pattern Recognit. 1996;29(7):1105–15.
16. Schwartz R. The pentagram map. Exp Math. 1992;1(1):71–81.
17. Su Z, Li L, Zhou X. Arc-length preserving curve deformation based on subdivision. J Comput Appl Math. 2006;195(1–2):172–81.
18. Tan W, et al. A novel approach to 2-D shape representation based on equilateral polygonal approximation. In: 2008 International conference on computer science and software engineering, vol 1. IEEE (2008).
19. Thompson R. Carleton College faculty website. http://people.carleton.edu/~rthompson
20. Valentin O, Schwartz R, Tabachnikov S. The pentagram map: a discrete integrable system. Commun Math Phys. 2010;299(2):409–46.