

Measuring Linearity of Ordered Point Sets

Milos Stojmenovic and Amiya Nayak

SITE, University of Ottawa,
Ottawa, Ontario, Canada K1N 6N5
{mstoj075, anayak}@site.uottawa.ca

Abstract. It is often practical to measure how linear a certain ordered set of points is. We are interested in linearity measures which are invariant to rotation, scaling, and translation. These linearity measures should also be calculated very quickly and be resistant to protrusions in the data set. No such measures exist in literature. We propose several such measures here: average sorted orientations, triangle sides ratio, and the product of a monotonicity measure and one of the existing measures for linearity of unordered point sets. The monotonicity measure is also a contribution here. All measures are tested on a set of 25 curves. Although they appear to be conceptually very different approaches, six monotonicity based measures appear mutually highly correlated (all correlations are over .93). Average sorted orientations and triangle side ratio appear as effectively different measures from them (correlations are about .8) and mutually relatively close (correlation .93). When compared to human measurements, the average sorted orientations and triangle side ratio methods prove themselves to be closest. We also apply our linearity measures to design new polygonal approximation algorithms for digital curves. We develop two polygonization algorithms: linear polygonization, and a binary search polygonization. Both methods search for the next break point with respect to a known starting point. The break point is decided by applying threshold tests based on a linearity measure.

Keywords: Linearity, ordered point sets, polygonization.

1 Introduction

The main motivation for this work is in image processing. Measuring the linearity of a finite set of points can become an interesting way of identifying the important components of a picture. Linear points often represent a region of interest in an image. By dissecting an object into an ordered collection of lines, the object becomes more easily identifiable; visually and computationally. Polygonization is the natural extension and cleanest application to measuring linearity. Simple objects are often a collection of straight lines. A square, triangle or even a star can be represented using a few vertices rather than a large number of points. For instance, polygonization is the basic technique of template matching in production facilities. Newly manufactured parts need to look like the master template to be acceptable. Polygonization is used to quickly compare the master to the copy.

In general, we are interested in measuring how linear a finite set of points is, and how well we can produce polygonizations based on these linearity measures. In analyzing various linearity algorithms, we align ourselves with the following criteria. We are interested in assigning linearity values to sets of points. The linearity value is a number in the range $[0, 1]$. It equals 1 if and only if the shape is linear, and equals 0 when the shape is circular or has another form which is highly non-linear such as a spiral. A shape's linearity value should be invariant under similarity transformations of the shape, such as scaling, rotation and translation. The algorithms should also be resistant to protrusions in the data set. Linearity values should also be computed by a simple and fast algorithm.

It is very important to stress that points in the set are ordered. This means that figures such as ellipses or rectangles which are very flat (long and thin) are considered to be highly nonlinear. If we were to consider unordered sets of points, such ellipses would be highly linear. The only concrete discussion on measuring linearity was found in [13], but it addresses unordered sets. In [13], six linearity measures of unordered sets were described.

Here, we will propose and analyze 8 algorithms that assign linearity values to ordered sets of points. The linearity algorithms are called: average sorted orientations, triangle sides ratio, and the last 6 deal with monotonicity multiplied by corresponding linearity measures from [13] for unordered point sets.

Average sorted orientations finds the unit vectors along the selected ordered pairs, and their average vector. The linearity measure is the length of that vector. *Triangle sides ratio* method takes random triplets of ordered points $A < B < C$, along a curve, and compares the side $|AC|$ to the sum $|AB| + |BC|$ of the other two sides.

We propose a measure of monotonicity of ordered point sets. *Monotonicity* finds the orientation line of the set of points and measures the behavior of the set with respect to this line. This is done by taking $O(N)$ pairs of points which are relatively close together along the curve and finding the vector which defines them, where N is the number of points along the curve. For each ordered pair $A_i B_i$, its sign is positive or negative depending on the angle (acute or obtuse) it makes with the fixed direction of the orientation line. The signed sum of lengths $|A_i B_i|$ is compared against the sum of lengths $|A_i B_i|$. Monotonicity is multiplied by each linearity measure for unordered sets to make combined metrics that measure the linearity of sorted point sets.

There are many publications that deal with polygonization, some of which are: [7, 8, 3, 9, 11, 14, and 15]. The algorithms need to be sensitive to large extrusions in the curve but should not react to small one which could be due to noise. They should not be restricted to modeling curves with fixed numbers of points, or with fixed lengths of line segments. Polygonization should be computed by elegant and fast algorithms. We also develop two polygonization algorithms which are called linear polygonization, and a binary search polygonization. Both polygonization methods try to split curves into appealing line segments based on a combined linearity and distance threshold. Linear polygonization corresponds to existing methods that successively add the next point from the set to the approximating line until that line violates the linearity test. We propose to use a $D(1-L) < T$ test, where D is the distance between endpoints of the candidate edge, L is the linearity measure, and T is the parameter threshold. Polygonization starts again from the last point that violated the threshold. Binary search polygonization is a novel approach. It looks for the longest

possible point segment that has a satisfactory linearity measure by applying a binary search to the other end of the next polygon edge. Thus a longer edge may emerge, leveraging out some local irregularities. The polygonization algorithms were tested using polygonization measures defined in [7, 8, S, 11]. Although they appear to be conceptually very different approaches, we show that most of the linearity algorithms provide similar polygonization results.

Overall in this paper, one monotonicity measure, 8 new linearity measures for ordered point sets and two new polygonization techniques are proposed. The literature review is given in section 2. Linearity and monotonicity measures and polygonal approximation algorithms are presented in section 3. The algorithms were tested on selected curves in section 4.

2 Literature Review

We will describe several well known functions on finite sets of points that are used in our linearity measures here. Existing linearity measures for unordered set will be covered along with other relevant measures. We also discuss existing polygonal approximation methods of digitized curves based on some linearity tests. Finally, we describe some existing measures of quality of polygonization.

2.1 Moments and Orientation

The central moment of order pq of a set of points Q is defined as:

$$\mu_{pq} = \sum_{x,y \in Q} (x - x_c)^p (y - y_c)^q,$$

where S is the number of points in the set Q , and (x_c, y_c) is the center of mass of the set Q . The center of mass is the average value of each coordinate in the set, and is determined as follows:

$$(x_c, y_c) = \left(\frac{1}{S} \sum x_i, \frac{1}{S} \sum y_i \right),$$

where (x_i, y_i) , $1 \leq i \leq S$, are real coordinates of points from Q . The angle of orientation of the set of points Q is determined by [1]:

$$angle = 0.5 \arctan \left(\frac{2\mu_{11}}{\mu_{20} - \mu_{02}} \right).$$

2.2 Linearity Measures for Unordered Sets

The most relevant and applicable shape measure to our work is the measuring of linearity of unordered data sets. [13] is the only source in literature that deals directly with measuring linearity for unordered sets of points. Six linearity measures were proposed in [13], all of which we will adapt here. The *average orientation* scheme first finds the orientation line of the set of points using moments. The method takes k pairs of points and finds the unit normals to the lines that they form. The unit normals all point in the same direction (along the normal to the orientation line). The average normal value (A, B) of all of the k pairs is found, and the linearity value is calculated

as $\sqrt{A^2 + B^2}$. *Triangle heights* takes an average value of the relative heights of triangles formed by taking random triplets of points. Relative heights are heights that are divided by the longest side of the triangle, then normalized so that we obtain a linearity value in the interval $[0, 1]$. *Triangle perimeters* takes the normalized, average value of the area divided by the square of the perimeter of triplets of points as its linearity measure. *Contour smoothness* and *eccentricity* were adapted from measures of circularity. They are simple formulas involving moments that were found in literature, and adapted to finding linearity [1]. The idea remained the same, but the resulting measurements were interpreted differently. In the original scheme in [1], they proposed a measure of circularity by dividing the area of a shape by the square of its perimeter. For circles, they arrived at circularities of 1, and values of less than 1 for other objects. *Ellipse axis ratio* is based on the minor/major axis ratio of the best ellipse that fits the set of points.

2.3 Relevant Measures

The standard method for measuring rectangularity is to use the ratio of the region's area against the area of its minimum bounding rectangle (MBR) [5]. A weakness of using the MBR is that it is very sensitive to protrusions from the region. A narrow spike out of the region can vastly inflate the area of the MBR, and thereby produce very poor rectangularity estimates. This goes against our stated criteria. Three new methods for measuring the rectangularity of regions are developed by Rosin [5].

Zunic and Rosin [16] described shape measures intended to describe the extent to which a closed polygon is rectilinear (each corner angle is 90° or 270°). The most frequently used convexity measure in practice is the ratio between the area of polygon and area of its convex hull [12]. Zunic and Rosin [17] discussed two measures that have advantages when measuring convexity of shapes with holes. Rosin [4] described several measures for sigmoidality.

Broder and Rosenfeld [BR] define a measure of collinearity merit for two adjacent line segment AB and CD ($A < B < C < D$ on the curve) as follows. The measure is $|AD|(|AB|+|CD|)/(|AB|+|BC|+|CD|)^2 \cos \angle ABC \cos \angle BCD$. However it cannot be used to measure colinearity of four arbitrary points from the curve. It also does not give colinearity close to 1 for digital lines due to frequent 'staircases' among pixels. In fact, it may even produce a zero factors in the product for certain straight line digitizations. Therefore we could not adopt it for our application.

2.4 Polygonal Approximation of Digitized Curves

There exist over a hundred algorithms for polygonization of digitized curves. We will discuss those that are based on similar strategy that is used here. A starting point S is selected, and the longest possible line segment SA . Then A becomes the new starting point (breakpoint) and the procedure repeats until the endpoint is reached. Methods differ based on the 'almost colinearity' test used. We identified only 'linear search' based methods, where colinearity is tested by successive adding one pixel at a time until the colinearity test fails.

Williams [W] used the cone intersection method to find the maximal possible line segment. Circles of fixed radius are drawn around each point. Points are then added one after another to the initial point until the intersection of all the cones with their vertex at the initial point and touching these circles is an empty set. The test used by Pavlidis [P] is based on maximal distance to a specific line segment and number of sign changes of points with respect to that line, and three parameters. Wall and Danniellsson [WD] find the maximal line segment by merging points one after another until the area deviation per unit length of the approximating line segment exceeds a maximum allowed threshold value. Ray and Ray [9] define three points to be collinear if the radius of largest inscribed circle inside their triangle is below a certain threshold. They assume that a set of points is collinear if the set containing the initial point and the last two points is collinear, and, recursively, the same set of points, without the last one, is collinear.

A common feature of all of the described methods is that they are based on a test (without measuring how linear a segment is) with parameters that are not scale invariant. Thus the same curve will yield different polygonizations on digitizations with different resolutions or with different curve sizes.

2.5 Evaluation of Polygonization Methods

Rosin [7, 8] identified enormous number of available methods for finding polygonal approximations, and discussed techniques for evaluating them. One popular metric is figure of merit, $FOM=N/(M*ISE)$ [S], where N is the number of points on the curve, M is the number of vertices of the obtained polygon, and ISE is the integral square error. ISE is the sum of squares of distances of each point on the curve from the corresponding edge of the polygon. [11] observes that ISE depends on the size of object, but does not change the definition. Marji and Siu [3] modified FOM by adding a parameter n , defining it as $(N/M)^n/ISE$, where $n=1,2,3$.

Rosin [7] proposed several new measures of polygonization quality: fidelity, efficiency and merit. *Fidelity* is measured as Eo/E , where E is ISE produced by measured algorithm, and Eo is ISE produced by the optimal algorithm, which is the one that produces the same number of lines but minimizes ISE . *Efficiency* is measured as M_o/M , where M is the number of lines produced by measured algorithm, and M_o is the number of lines produced by an optimal algorithm, which is the minimal possible number of lines of an algorithm that produces the same ISE . *Merit* is then defined as $\sqrt{fidelity * efficiency}$.

Rosin [8] proposed *monotonicity* and *consistency* measures of polygonization stability. *Monotonicity* measures how well the number of generated lines and ISE by given method follow selected parameter in the given method. In a number of domination point based methods, the selected parameter is M , the desired number of lines. A perfectly monotone method is then supposed to always reduce ISE when M is increased, and *monotonicity* measures the amount of discrepancy in such behavior. *Consistency* measures the effect of deleting certain number of pixel from the beginning of the curve on breakpoint position, number of breakpoints, ISE , merit etc.

3 Measuring Linearity and Polygonization

Here, we will describe the measures of linearity of ordered sets, the monotonicity measure, and the two polygonization schemes which constitute the contributions of this article. The monotonicity measure is the only one not used independently in either linearity measurement or polygonization. It is combined with other measures to achieve usable results.

3.1 Linearity of Ordered Sets of Points and Monotonicity

A. Average Sorted Orientations

Average sorted orientations works much the same as its average orientations predecessor [13], but points are taken such that the second comes after the first in the ordered set. Also, the slopes are directly averaged, without any reference to the orientation line. Figure 1 shows that only the magnitude of the average ordered unit vectors determines the linearity measure. This makes this metric faster to compute than the average orientations protocol in [13].



Fig. 1. Average Sorted Orientations

B. Triangle Sides Ratio

Here, we take k ordered triplets of points, and measure their average colinearity. For each triple, divide the length of the side of the triangle whose endpoints are furthest apart in the sorted point array by the sum of the lengths of the other two sides. Figure 2 shows that the red side of the triangle AC is the one whose endpoints are furthest apart in the ordered array. The length of the side AC is divided by the sum of the lengths of the other two blue sides.

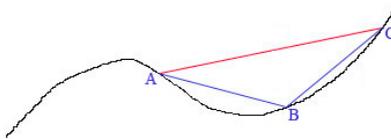


Fig. 2. Triangle sides ratio example

C. Monotonicity

Monotonicity measures the behavior of curves with respect to their orientation line. It is expected that monotonic curves define a more linear ordered set of points than non-monotonic curves. Here, we first find the orientation line of the set of point which we are interested in, and measure the monotonicity of this set relative to this line. Since

the moment function sometimes produces orientation lines which are 90 degrees offset from what is visually the actual orientation line, we repeat the entire procedure for $angle=angle+\pi/2$, and select the higher of the two measures. In Figure 3, we see that the curve on the left is completely monotone with respect to its blue orientation line, whereas the curve on the right is not monotone.

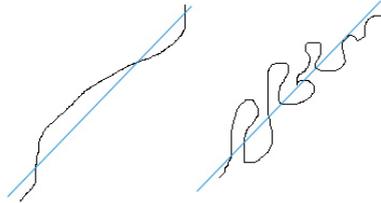


Fig. 3. Monotonicity examples

The algorithm works by taking $N-4$ pairs of points which are 4 positions apart. We chose to take points which are 4 positions apart in the array of ordered points since points which were closer together did not represent the natural slope of the curve due to digitization. A vector v is found for each pair of points. For each v , we multiply it by the orientation line vector of the whole set of points via a dot product. If the dot product is positive, the sign s of the magnitude mag of v is positive, otherwise it is negative. The sum of all $s*mag$ is divided by the sum of the absolute values of all mag to form a monotonicity value. Monotonicity is multiplied by each linearity measure to make combined metrics that measure the linearity of sorted point sets.

Figure 4 shows a concrete yet simplified example of how the monotonicity algorithm works. We take only 4 pairs of points to capture the spirit of the method. The orientation line is seen below the figure. All of the dark blue arrows show pairs of points. The only pair that will have a negative dot product with the orientation line is pair A_5A_6 . The monotonicity value for these 4 pairs is (absolute value of):

$(|A_1A_2|+|A_3A_4|-|A_5A_6|+|A_7A_8|)/(|A_1A_2|+|A_3A_4|+|A_5A_6|+|A_7A_8|)$. Note that the light blue orientation line should actually be paced right over the curve, but was shifted to the side to avoid complicating the figure.

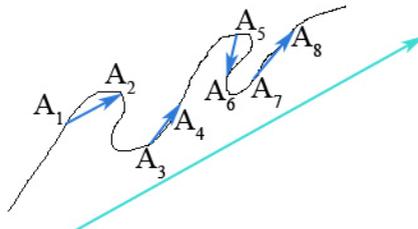


Fig. 4. Monotonicity functional example

The ordered linearity values of a shape are the result of the monotonicity measure of that shape multiplied with one of the six linearity measures from [13] for unordered data.

3.2 Polygonal Approximation of Curves

Both polygonization algorithms are designed to be invariant to rotation. This is achieved through a mechanism that always selecting the same starting point. The starting point is chosen such that it encompasses the longest linear segment that does not violate the linearity threshold of that curve.

A. Linear Extension

This method uses a brute force method of polygonization by comparing the linearity of growing line segments to a threshold which depends on the length of the line segment and a linearity value. The threshold is in the form $D(1-L) < T$, where D is the length of the line segment, L is its linearity, and T is the threshold value. The point at which the threshold is violated becomes a new polygonization point, and the process is continued from this new point.

Algorithm: *linear extension*:

Input: array of points: $Points=(X_i, Y_i), 1 \leq i \leq N$, threshold T ;

Output: *set of points which define a polygonal approximation*;

Find longest segment A_1A_2 such that linearity of $A_1A_2 \leq T$, make A_1 starting point of polygonization;

```

For  $i=1$  to  $count$  do {
     $tempLinearity=1$ ;  $pointsToTry=4$ ;
    while ( $tempLinearity \geq T$ ) and ( $i+pointsToTry < N$ ) do {
         $tempLinearity= Distance * Linearity$  (both between  $points[i]$ , and
             $points[i+pointsToTry]$ );
         $pointsToTry++$ ;
    }
    Memorize point:  $Points[i+pointsToTry]$  as one of points of approximating
        polygon;
     $i = i+pointsToTry-1$ ;
}

```

B. Binary Search Extension

This is an improvement over the linear search polygonization method of the previous procedure. The problem with the first procedure is that new lines can be locally selected too quickly, while a generally long line may exist which can approximate the curve with fewer line segments. Given that we are performing a binary search, this algorithm performs faster than the linear extension method.

Algorithm: *Binary Search Extension*:

Input: array of points: $Points=(X_i, Y_i), 1 \leq i \leq N$, threshold T ;

Output: *set of points which define a polygonal approximation*;

Find longest segment A_1A_2 such that linearity of $A_1A_2 \leq T$, make A_1 starting point of polygonization;

```

int start=1;
While start<>N do {
    start=polygonize(start, N, T, Points);
    memorize points[start] as next polygonization point;
}

```

```

Procedure polygonize(start, end, T, Points){
    if (start=end) then return start;
    mid=(start+end)/2;
    find linearity lin of line segment start to mid by one of
        the linearity methods;
    find monotonicity mon of segment start to mid;
    tempLinearity=ecc*mon;
    dis=Euclidian distance between points start and mid;
    templinearity=dis*(1- tempLinearity);
    if (templinearity ≤ T) {
        start=polygonize(mid, end, Points);
    } else {
        start=polygonize(start, mid, Points);
    }
    return start;
}

```

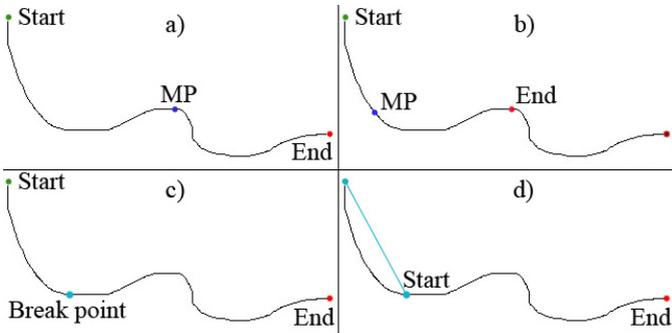


Fig. 5. Binary polygonization process

The whole process is seen in Figure 5. In Fig. 5a), we see that the starting point is the green dot labeled *start*, the ending point is the red dot labeled *end*, and the midpoint between them is represented by a blue dot, and labeled *mp*. The algorithm first checks the linearity value of segment start to *MP*. If this linearity value satisfies the threshold, the next breakpoint is sought after in the segment *MP* to *end*. In this example, we see that the segment *Start* to *MP* is not linear enough, and therefore, the binary algorithm further searches for a breakpoint here. This is seen in Figure 5b), where the new *mid* and *end* points are marked. By continuing the binary search, we find the breakpoint in Figure 5c). The segment *Start* to *Break point* is stored as part of the polygonization. The polygonization algorithm will restart from this breakpoint in Fig. 5d).

4 Experimental Data

The testing of the algorithms was done on two sets. The linearity algorithms were tested on a set of 25 non trivial shapes, shown in Figure 6. These shapes were assembled by hand and are meant to cover a wide variety of non trivial curves. The polygonization algorithms were executed on a set of 21 standard test curves found in [8], shown in Figure 7. Each curve comprises between 100 and 500 points.

Several linearity measurement algorithms are based on selecting k pairs or triples of points from the set. When such procedure is called frequently by linear search based polygonization algorithm, the speed may become a concern. We first added speedup procedures to determine sample size k so that linearity measure has sufficient accuracy. In application, such as polygonization, it may be more common to make a judgment on whether or not the set is linear. This provided further speedup. Both speed improvements were made using basic probability theory for confidence intervals [2].

Table 1 shows the linearity testing results for the Average Sorted Orientations (ASO), and Triangle Sides Ratio (TSR) algorithms. It also shows linear measures for six methods AO, EC, TH, TR, CS, and EAR. They are proposed in [13] for unsorted data sets, and were multiplied by the monotonicity values (using the algorithm described in this article) of each curve to make them relative to sorted data sets which are relevant here. The monotonicity values (MON) of each curve are seen in the last column.

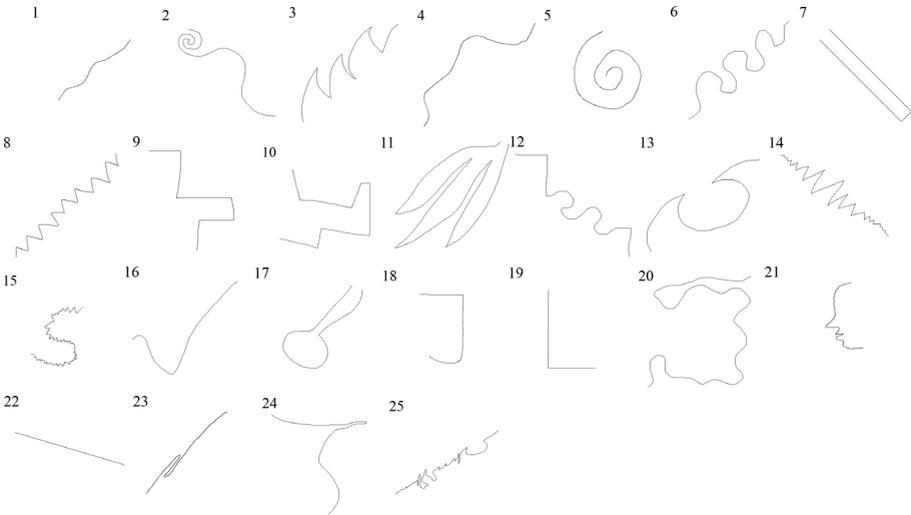


Fig. 6. Linearity test set

To achieve higher consistency (as defined by [8]) for both polygonization methods, and therefore rotation independency, the initial point has been selected by first searching for the longest line segment in the polygon. Starting from each point, linearity measure was applied by linear search, extending the line as far as possible. The starting point of the longest such line segment was selected as the initial point.

Visual inspection of the polygonization results of the binary polygonization method can be seen in Figure 8. The binary search method was used here. The only parameter in this polygonization run is the *linearity*distance* threshold, which was set to $T=1$, which provided similar values for M as those seen in cited references. The linearity measure that was used to generate the polygonization was the eccentricity (EC) measure. The light red dot on each curve represents the starting point of each algorithm.

Tables 2 and 3 show the polygonization results of linear extension algorithm as it is tested on the curves in Figure 7. Table 2 shows the number of points in each polygonization, and Table 3 shows the measure of $N^3/(M^3*ISE)$, where N is the number of points in the original curve, and M is the number of points in the approximation. This measure is taken from [3] for $n=3$. It is selected for the following reason.

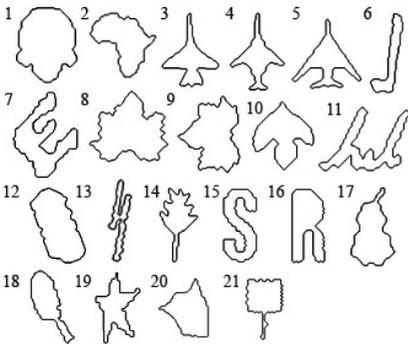


Fig. 7. Polygonization set

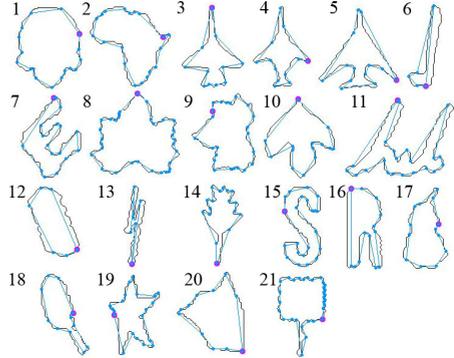


Fig. 8. Binary Polygonization

Table 1. Linearity Results

	AO	EC	TH	TP	CS	EAR	ASO	TSR	MON
1	.95	.99	.85	.97	.80	.93	.98	.99	1
2	.35	.56	.29	.37	.23	.47	.71	.71	.65
3	.34	.47	.25	.31	.20	.40	.85	.91	.54
4	.71	.83	.61	.79	.50	.70	.90	.92	.98
5	.01	.02	0	0	0	.02	.27	.18	.16
6	.36	.55	.24	.32	.19	.46	.82	.90	.65
7	.10	.12	.07	.08	.06	.10	.26	.49	.13
8	.83	.96	.73	.85	.66	.88	.94	.95	.98
9	.35	.47	.25	.31	.20	.39	.81	.83	.69
10	.04	.06	.02	.03	.02	.05	.45	.59	.20
11	.05	.09	.02	.03	.01	.08	.47	.25	.16
12	.61	.79	.50	.64	.42	.68	.89	.92	.87
13	.23	.36	.13	.17	.11	.29	.71	.79	.53
14	.57	.71	.44	.53	.38	.63	.91	.95	.75
15	.09	.16	.07	.08	.05	.14	.70	.71	.32
16	.35	.53	.35	.39	.32	.44	.79	.90	.70
17	.05	.07	.03	.04	.02	.05	.25	.26	.09
18	.26	.43	.28	.36	.22	.35	.76	.69	.65
19	.59	.73	.56	.68	.49	.61	.85	.91	1
20	.08	.13	.06	.08	.04	.11	.61	.50	.52
21	.44	.62	.42	.54	.34	.52	.86	.82	.80
22	1	1	.99	1	.99	1	1	1	1
23	.64	.66	.56	.61	.53	.63	.85	.96	.67
24	.04	.22	.24	.28	.22	.20	.64	.69	.78
25	.41	.50	.36	.42	.32	.45	.91	.97	.51

Table 2. Linear Polygonization-Break Points

	N	AO	EC	TH	TP	CS	EAR	ASO	TSR
1	366	43	18	28	22	29	34	17	31
2	413	55	20	36	35	36	48	21	40
3	377	41	12	28	17	29	28	13	31
4	352	39	13	27	19	28	36	13	35
5	482	59	13	39	22	39	46	16	40
6	317	22	6	23	13	23	27	9	25
7	484	53	18	39	35	41	52	25	44
8	474	66	33	43	44	43	60	31	37
9	384	48	24	34	32	34	45	23	36
10	372	40	15	27	19	27	33	15	31
11	674	73	23	54	43	56	70	29	59
12	290	34	8	27	18	28	34	13	30
13	376	35	7	35	28	36	40	14	29
14	366	40	18	30	28	30	37	17	35
15	483	56	23	38	35	38	49	26	52
16	427	29	14	21	20	22	27	15	33
17	329	38	14	29	25	29	40	16	32
18	326	34	13	27	23	27	37	16	33
19	450	49	21	35	35	35	45	23	38
20	386	46	12	30	29	30	39	16	34
21	419	50	32	33	37	34	48	27	40
av	407	45	17	33	28	33	41.7	19	36

Our goal is to design and evaluate polygonization methods that are invariant to curve scaling and/or resolution applied. The curve size is described by a single parameter N , the number of pixels in a given digitization. Distances are measured in this ‘pixel’ geometry, where the distance between two neighboring pixels in horizontal/vertical direction is 1.

Consider two digitizations of the same curve, with $N_1=N$ and $N_2=mN$ pixels, respectively. Assuming that polygonization is invariant to scaling, they should yield the same M , the number of selected polygon vertices, and these vertices should be at

Table 3. Linear polygonization measure $n3/(m3*ISE)$

	N	AO	EC	TH	TP	CS	EAR	ASO	TSR
1	366	1.42	3.08	1.07	1.06	1.06	0.94	2.41	.00
2	413	0.89	1.93	1.43	1.29	1.43	0.16	2.34	.45
3	377	0.34	2.72	0.00	1.02	0.00	0.85	3.31	0
4	352	0.99	1.64	2.44	0.70	2.57	1.80	3.07	.23
5	482	0.09	1.10	2.09	1.75	2.13	0.00	0.00	.18
6	317	3.51	2.32	3.72	6.96	3.78	0.00	4.33	0
7	484	0.90	3.46	1.74	1.40	1.46	0.00	1.93	.27
8	474	0.00	0.92	0.00	0.66	0.91	0.00	1.00	.39
9	384	0.68	1.29	1.28	0.44	1.36	0.82	1.33	.15
10	372	1.55	1.72	2.05	1.40	1.93	0.00	3.95	.31
11	674	0.69	1.80	1.13	1.30	1.09	0.00	1.47	.15
12	290	1.89	3.87	4.38	3.26	2.86	0.00	4.52	.45
13	376	2.12	5.52	1.74	2.20	0.00	2.04	5.27	.41
14	366	1.45	1.52	1.14	2.27	1.09	1.29	0.35	.32
15	483	0.66	0.64	1.68	0.97	1.60	0.97	1.31	.21
16	427	0.85	2.67	2.90	1.79	2.81	1.90	2.93	0
17	329	1.02	2.63	1.19	1.94	1.19	1.18	1.92	.30
18	326	0.93	2.67	3.03	2.44	2.92	1.23	0.00	.86
19	450	0.59	0.82	1.84	0.00	1.80	0.54	2.25	0
20	386	0.97	3.19	3.28	1.72	3.14	1.49	5.14	.36
21	419	0.81	1.76	1.40	1.05	0.00	0.88	1.45	0
av	407	1.06	2.25	1.88	1.70	1.67	0.77	2.39	.24

positions with respect to the curve. The distances of corresponding pixels in two digitizations are then e and me , respectively. Then $ISE_2 \approx \sum_i^{mN} (me)^2 \approx m^3 ISE_1$. Thus $N_2^3/(M^3 ISE_2) \approx (mN)^3/(M^3 m^3 ISE_1) = N_1^3/(M^3 ISE_1)$.

This means that the measure $N^3/(M^3 ISE)$ is invariant to curve scaling or resolution, for any t . Similar analysis can also show that some popularly used evaluation measures are not scalable, and that the evaluation measures depend on the numbers of pixels taken in a particular digitization. Measures which are not invariant to the

Table 4. Binary polygonization-break points

	N	AO	EC	TH	TP	CS	EAR	ASO	TSR
1	366	51	18	29	26	29	40	18	42
2	413	81	20	36	34	36	54	21	53
3	377	48	12	25	24	29	36	18	43
4	352	53	13	25	24	30	40	16	55
5	482	68	13	42	28	42	44	22	60
6	317	33	6	26	15	31	33	11	38
7	484	79	18	44	38	40	60	25	58
8	474	87	33	46	50	45	71	35	53
9	384	68	24	30	40	34	54	25	51
10	372	51	15	33	23	33	38	20	43
11	674	94	23	70	46	73	79	31	79
12	290	45	8	24	25	28	39	14	39
13	376	39	7	32	30	37	49	18	38
14	366	55	18	35	27	35	49	21	53
15	483	82	23	38	39	39	64	26	70
16	427	38	14	25	26	26	38	17	44
17	329	46	14	32	25	32	45	15	43
18	326	50	13	29	28	30	40	17	46
19	450	71	21	33	38	41	55	21	55
20	386	61	12	27	26	31	46	14	48
21	419	80	32	39	46	38	58	29	52
av	407	61	17	34	31	36	49.1	21	51

Table 5. Binary polygonization measure $n^3/(m^3 * ISE)$

	N	AO	EC	TH	TP	CS	EAR	ASO	TSR
1	366	0.11	3.08	2.40	1.27	0.95	1.48	2.13	0.37
2	413	0.32	1.93	0.70	1.85	0.83	1.46	1.96	0.17
3	377	0.09	2.72	1.54	0.43	0.52	1.81	1.59	0.14
4	352	0.37	1.64	0.20	1.13	1.08	1.44	0.96	0.17
5	482	0.05	1.10	0.41	0.89	0.33	1.36	1.99	0.19
6	317	0.11	2.32	0.60	2.59	1.49	4.24	0.73	0.59
7	484	0.20	3.46	0.31	1.07	0.13	1.21	1.98	0.40
8	474	0.28	0.92	0.30	0.64	0.38	0.50	0.84	0.17
9	384	0.21	1.29	0.19	0.53	0.67	0.68	0.73	0.19
10	372	0.46	1.72	0.88	2.21	0.48	1.07	0.48	0.25
11	674	0.41	1.80	0.28	1.21	0.35	0.35	0.71	0.31
12	290	0.69	3.87	1.52	1.26	1.65	1.75	0.95	0.41
13	376	0.43	5.52	0.10	0.81	0.28	0.72	1.56	0.18
14	366	0.17	1.52	0.58	0.57	0.39	0.50	1.06	0.72
15	483	0.20	0.64	0.58	1.80	0.54	1.03	1.61	0.44
16	427	0.67	2.67	2.25	1.83	2.03	0.76	6.20	0.09
17	329	0.75	2.63	0.21	1.12	0.06	2.15	2.95	0.61
18	326	0.27	2.67	1.90	0.82	1.76	0.90	1.92	0.46
19	450	0.19	0.82	0.26	1.05	0.97	0.36	1.39	0.29
20	386	0.09	3.19	0.20	0.42	1.65	0.15	3.98	0.56
21	419	0.20	1.76	0.77	0.47	0.79	0.47	1.21	0.28
av	407	0.30	2.25	0.77	1.14	0.82	1.16	1.76	0.33

number of pixels on the curve include FOM [10] used very frequently (it behaves as the inverse of m^2), and $(N/M)^n/ISE$ [3] for $n \neq 3$.

Tables 4 and 5 show the binary polygonization results of this algorithm as it is tested on the curves in Figure 7. They follow the format and meaning of the previous two tables. The best algorithm for polygonization according to the $n^3/(m^3 * ISE)$ measure was the linear polygonization method combined with the average sorted orientations linearity metric. The closest competitor was binary polygonization combined with the eccentricity linearity metric.

5 Conclusion

Since the values N were not reported in given references, and the measures used were either not scalable or requiring further information about parameters used in testing, direct comparison of our polygonization schemes with other existing methods was left for future work. This is part of our plan for further extensions of this work in several directions. We will study new linearity measures, and new evaluation measures for polygonization, including new ways to evaluate *ISE*. We will then be in a position to properly compare some of the existing methods and our new ones, using scale invariant measures.

References

1. Csetverikov, D.: Basic algorithms for digital image analysis, Course, Institute of Informatics, Eotvos Lorand University, visual.ipan.sztaki.hu
2. Hogg, R.V., Tanis, E.A.: Probability and Statistical Inference. Prentice Hall, Englewood Cliffs (1997)
3. Marji, M., Siy, P.: Polygonal representation of digital planar curves through dominant point detection—a nonparametric algorithm. *Pattern Recognition* 37, 2113–2130 (2004)
4. Rosin, P.: Measuring sigmoidality. In: Petkov, N., Westenberg, M.A. (eds.) CAIP 2003. LNCS, vol. 2756, pp. 410–417. Springer, Heidelberg (2003)
5. Rosin, P.: Measuring Rectangularity. *Machine Vision and Applications* 11, 191–196 (1999)
6. Rosin, P.: Measuring shape: ellipticity, rectangularity, and triangularity. *Machine Vision and Applications* 14, 172–184 (2003)
7. Rosin, P.: Techniques for assessing polygonal approximations of curves. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 19(6), 659–666 (1997)
8. Rosin, P.: Assessing the behavior of polygonal approximation algorithms. *Pattern Recognition* 36, 505–518 (2003)
9. Ray, B., Ray, S.: An algorithm for polygonal approximation of digitized curves. *Pattern Recognition Letters* 13, 489–496 (1992)
10. Sarkar, D.: A simple algorithm for detection of significant vertices for polygonal approximation of chain-coded curve. *Pattern Recognition Letters* 14, 959–964 (1993)
11. Sarfraz, M., Asim, M.R., Masood, A.: Piecewise polygonal approximation of digital curves. In: Proceedings of the Eighth International Conference on Information Visualization, vol. 14, pp. 991–996 (2004)

12. Sonka, M., Hlavac, V., Boyle, R.: *Image Processing, Analysis, and Machine Vision*, Chapman & Hall (1993)
13. Stojmenovic, M., Nayak, A., Zunic, J.: Measuring linearity of a finite set of points. In: *IEEE International Conference on Cybernetics and Intelligent Systems (CIS)*, Bangkok, Thailand, June 7-9, pp. 222–227 (2006)
14. Teh, C., Chin, R.: On the detection of dominant points on digital curves. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 11(8), 859–872 (1989)
15. Ventura, J., Chen, J.M.: Segmentation of two-dimensional curve contours. *Pattern Recognition* 25(10), 1129–1140 (1992)
16. Zunic, J., Rosin, P.: Rectilinearity Measurements for Polygons. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 25(9), 1193–1200 (2003)
17. Zunic, J., Rosin, P.: A new convexity measure for polygons. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 26(7), 923–934 (2004)