Available online at www.sciencedirect.com
ScienceDirect
Pattern Recognition 41 (2008) 2503–2511
PATTERN RECOGNITION
THE JOURNAL OF THE PATTERN RECOGNITION SOCIETY
www.elsevier.com/locate/pr

Measuring linearity of planar point sets

Milos Stojmenović[a,*], Amiya Nayak[a], Jovisa Zunic[b]

[a]SITE, University of Ottawa, Ottawa, Ont., Canada K1N 6N5
[b]SECH, University of Exeter, Exeter EX4 4QF, UK

Received 17 April 2006; accepted 16 January 2008

Abstract

Our goal is to design algorithms that give a linearity measure for planar point sets. There is no explicit discussion on linearity in literature, although some existing shape measures may be adapted. 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. The measures of eccentricity and contour smoothness were adapted from literature, the other five being triangle heights, triangle perimeters, rotation correlation, average orientations, and ellipse axis ratio. The algorithms are tested on 30 sample curves and the results are compared against the linear classifications of these curves by human subjects. It is found that humans and computers typically easily identify sets of points that are clearly linear, and sets of points that are clearly not linear. They have trouble measuring sets of points which are in the gray area in-between. Although they appear to be conceptually very different approaches, we prove, theoretically and experimentally, that eccentricity and rotation correlation yield exactly the same linearity measurements. They however provide results which are furthest from human measurements. The average orientations method provides the closest results to human perception, while the other algorithms proved themselves to be very competitive.

Keywords: Linearity; Finite point sets; Moments

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 are interesting since they often represent a region of interest in an image. Most man made structures or objects have strong straight lines that are easily identifiable. By dissecting an object into an ordered collection of lines, the object becomes more easily identifiable; visually and computationally. There are a variety of methods to extract edges from images. Objects such as cars or tables in such edge representations of images are still easily recognizable by humans—the whole is more than just the sum of its parts (edges in our case). This leads to interesting possibilities for the domain of computer vision in the sense that useful information can be extracted from images just by examining the edges.

Here, we are interested in measuring linearity of a finite set of points in a plane. In analyzing various algorithms, we align ourselves with the following criteria. We are interested in assigning linearity values to sets of points. The linearity value is preferred to be a number from [0, 1] if such a normalization is possible. Naturally, it is preferred that the highest possible measured linearity correspond to a perfectly linear shape—i.e. to the sets whose points belong to a line. The linearity value of a given shape equals 1 if and only if the shape is linear, and the linearity value 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. Linearity values should also be computed by a simple and fast algorithm.

It is very important to stress that points in the set are not ordered. This means that curves such as ellipses or rectangles which are very flat (long and thin) are considered to be highly linear. If we were to consider ordered sets of points, such ellipses would be highly nonlinear. Because the set of points is not ordered, permutations of the input set should not affect the linearity value.

* Corresponding author. Tel.: +1 613 822 2554; fax: +1 613 822 0057.
E-mail addresses: mstoj075@site.uottawa.ca (M. Stojmenović), anayak@site.uottawa.ca (A. Nayak), J.Zunic@exeter.ac.uk (J. Zunic).

0031-3203/$30.00 © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2008.01.013

The closest applications of shape analysis to our article in this field are measuring convexity [1–4], rectilinearity [5], rectangularity [6,7], ellipticity [7], sigmoidality [8] and circularity [9]. We have not found a concrete discussion on measuring linearity. Some sources in literature make references to measurements that could be used to test linearity, but they use them for other purposes. For example, the measures of eccentricity and contour smoothness are developed for testing circularity, yet they are also used here for testing linearity. Measuring convexity is a popular problem, but one that cannot easily be applied here. Rectilinearity was discussed in literature and can be applied to finding man made settlements in satellite imagery [5]. Some rectangularity measures may be modified to measure linearity. Rectangles which are long, yet narrow, may represent lines. Some algorithms which measure rectangularity are sensitive to protrusions in the data set. For example, a smallest enclosing box can be used to measure rectangularity [6]; however, small irregularities in the data set can seriously affect the performance of this metric.

Here, we will propose and analyze several algorithms that assign linearity values to sets of points. They are called: average orientations, rotation correlation, triangle heights, triangle perimeters, contour smoothness, eccentricity, and ellipse axis ratio. Contour smoothness and eccentricity were adapted from measures of circularity.

The rotation correlation and average orientation schemes first find the orientation line of the set of points using moments. The average orientations 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 $\sqrt{A^2 + B^2}$. In the rotation correlation method, the set of points is then rotated such that its orientation line is 45° from the x-axis. This rotation is performed to give equal weights to both the x and y coordinate values in the correlation formula. Correlation is performed on the rotated set of points to determine linearity. Triangle heights takes an average value of the relative heights of triangles formed by taking random triplets of points, normalized so that we obtain a linearity value in the interval [0, 1]. Relative heights are heights that are divided by the corresponding longest side of the triangle. '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 are simple formulas involving moments that were found in literature [9], and adapted here to finding linearity. We prove, theoretically and experimentally, that the eccentricity and rotation correlation methods give same linearity measures. Ellipse axis ratio is based on the minor/major axis ratio of the best ellipse that fits the set of points.

The literature review is given in Section 2. Linearity measures are presented in Section 3. The results of the algorithms along with the comparison to the linearity classification of the shapes by humans are presented in Section 4. The algorithms were tested on a set of 30 shapes. These shapes were assembled by hand and are meant to cover a wide variety of non-trivial curves. The most interesting finding is that the rotation correlation method produces identical results on the set of test shapes to the eccentricity method. These two linearity measures are conceptually completely different, yet we show here that they are in fact the same measure. The closest one to human perception was the measure obtained from the contour smoothness algorithm. The methods based on sampling k pairs or triplets were faster than others, yet gave reasonably accurate linearity measures. Linearity values for large values of k gave similar results to linearity values for reasonable values of k such as 250–500.

2. Literature review

We will describe several well known functions on finite sets of points that are used in our linearity measures here.

2.1. Discussion on geometric moments for point sets, orientation and correlation

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), $1 \leqslant i \leqslant S$, are real coordinates of points from Q. The angle of orientation of the set of points Q is determined by [9]

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

The orientation line (also called the 'axis of the last second moment of inertia') is a line that minimizes the sum of squares of distances from points in the set. It is well known that the orientation line passes through the origin, and its slope is determined by the above given *angle*. Since $angle + \pi/2$ also satisfies the same equation, algorithms based on this formula need to verify two candidate orientation lines.

All definitions are applied on a set of points with real coordinates. In our examples, we use the moment calculations on just the finite set of points located on a closed or open curve.

We find that the orientation of the border points of a closed curve is almost identical to the orientation of all of the digital points inside the closed curve. We are especially interested in digitized curves because they are used in our experiments.

Correlation is another well known function that is used in this article. It returns a correlation value between −1 and 1:

$$correlation = \frac{\sum_{i=0}^{count} x_i y_i - \frac{\sum_{i=1}^{count} x_i \sum_{i=1}^{count} y_i}{count}}{\sqrt{\sum_{i=1}^{count} x_i^2 - \frac{(\sum_{i=1}^{count} x_i)^2}{count}}\sqrt{\sum_{i=1}^{count} y_i^2 - \frac{(\sum_{i=1}^{count} y_i)^2}{count}}}.$$

2.2. Relevant shape measures

The most relevant and applicable shape measure to our work is the measuring of rectangularity. 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) [6]. 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 [6]. They are tested together with the standard MBR method on synthetic and real data. It is concluded that, while all the methods have their drawbacks, the best two are the bounding rectangle and discrepancy methods. The discrepancy method estimates rectangle sides in two ways, and measures the agreement between the two. One of the ways to measure the sides of a rectangle is to find the best ellipse that corresponds to the region, and estimate the rectangle's measurements by using the minor and major axes of such an ellipse. This method uses second order moments. The formulas for the major axis a and minor axis b of the best fit ellipse are

$$a = \sqrt{2[\mu_{20} + \mu_{02} + \sqrt{(\mu_{20} - \mu_{02})^2 + 4\mu_{11}^2}]/\mu_{00}},$$

$$b = \sqrt{2[\mu_{20} + \mu_{02} - \sqrt{(\mu_{20} - \mu_{02})^2 + 4\mu_{11}^2}]/\mu_{00}}.$$

Zunic and Rosin [5] described shape measures intended to describe the extent to which a closed polygon is rectilinear (each corner angle is 90° or 270°). The two measures proposed in Ref. [5] are based on the maximum ratio of perimeters measured by the two metrics. One metric is the Euclidean distance while the other is the city block distance (sum of differences in each coordinate). When a polygon rotates, the city block-based perimeter changes. They prove that a polygon is rectilinear if and only if there exists an angle α such that the city block-based perimeter for the polygon rotated by α is the same as the Euclidean distance based perimeter. They show that these maximums for n-gons can be obtained by testing at most $4n$ angles of rotation.

The most frequently used convexity measure in practice is the ratio between the area of a polygon and area of its convex hull [1]. Zunic and Rosin [2] discussed two measures that have advantages when measuring convexity of shapes with holes. Zunic and Rosin [2] first proposed to measure the ratio of the largest convex polygon contained inside a given one, and the area of that polygon, but noted that it is computationally expensive to apply. Then they proposed to measure the ratio of the Euclidean perimeter of a given shape and the Euclidean perimeter of its convex hull. Convexity measures were also studied in Refs. [3,4].

Rosin [8] described several measures for sigmoidality. It is roughly a measure of the 'S' shape where the 'fullness' (thickness) of the shape is not taken into account. Rosin [8] proposed to fit cubic polynomials, but without the quadratic term, to ensure a symmetric curve. Data are rotated so that principal axis is the x-axis, then least square fitting is applied. The correlation coefficient is used to measure the quality of the fit. Negative correlations are ignored, so that the interval is [0, 1]. The second approach in Ref. [8] is to consider tangent angles, which look somewhat Gaussian. The function parameters are determined by matching mean absolute values and variances, the area under the curve is normalized to one, and correlation is used to measure sigmoidality. The third approach in Ref. [8] is based on curvature analysis. Positive and negative curvature values are separated and summed over the curve to the left and right of midpoint. The sums should be large and differences small with respect to overall area of the curve with respect to the central line of symmetry.

The contour smoothness measure was described in Ref. [9] as a measure of circularity, and was adapted and converted here into a measure of linearity. The idea remained the same, but the resulting measurements were interpreted differently. In the original scheme in Ref. [9], 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.

3. Measuring linearity

The algorithms that we proposed and analyzed for measuring linearity are described here. They are called: average orientations, rotation correlation or eccentricity, triangle heights, triangle perimeters, contour smoothness, and ellipse axis ratio. All of the algorithms give results which are invariant to scaling, translation and rotation of sets of points. The average orientation, triangle heights and triangle perimeters algorithms use a parameter k which represents the sampling rate of points taken to determine linearity. This k can be automatically determined by each algorithm, for a sufficiently accurate linearity measure, or for rejecting linearity of a set with sufficient confidence.

3.1. Average orientations

Here, we first find the center of mass of the point set, and its angle of orientation using moments. This function takes k random pairs of points along the curve. It finds their slopes (m), and finds the normals to their slopes $(-m, 1)$. Each normal is saved as a vector $(-m/norm, 1/norm)$ in array ab, where $norm = \sqrt{m^2+1}$ is a normalization factor. These vectors are compared against the normal to the orientation line

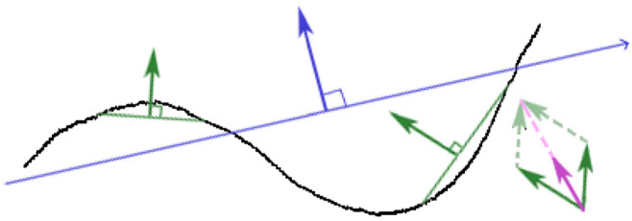

Fig. 1. Normals all oriented in the same direction.

determined by the moments formula above $(-M, 1)$, where $M = \tan(angle)$. The dot product of $(-M, 1)$ and $(-m, 1)$, for each pair of points is evaluated as $dp = mM + 1$. If $dp < 0$, the vector $(m/norm, -1/norm)$ is stored instead. All normals are oriented to point in the same general direction with respect to the vector $(-M, 1)$. They are pointed in the same direction since the vectors would otherwise cancel each other out in the case of a perfectly straight line, and give a linearity value near 0, see Fig. 1.

These normals in array *ab* are averaged out, and the resulting normal (A, B) is deemed to be the normal to the orientation of the curve. The averaging is done separately for each vector coordinate. The measure of linearity is defined as $\sqrt{A^2 + B^2}$. In the case of a perfectly straight line, all of the unit vectors would point in the same direction, and have a height of 1 with respect to the orientation line. Otherwise, the resulting average orientation would not be orthogonal to the orientation line, and would have a magnitude less than 1.

Algorithm. *Average orientations.*

Input: array of points: $Points = (X_i, Y_i),\ 1 \leqslant i \leqslant count,\ k$;
Output: *linearity* value;

Array a, b; //normals to each line found in loop
Find center of mass (x_c, y_c) using moments;
Find slope $M \leftarrow \tan(angle)$ of orientation of curve
Vector $(-M, 1)$ is the normal to this line;

For $i = 1$ **to** k **do** {
Take two random points (x_1, y_1), (x_2, y_2) from *Points*;
Find slope between them, call it $m = (y_2 - y_1)/(x_2 - x_1)$;
$a1 \leftarrow -m$, $b1 \leftarrow 1$;
$dp \leftarrow mM + 1$;
Normalization factor $norm \leftarrow \sqrt{(m^2 + 1)}$;
If $(dp < 0)$ **then** $\{a[i] \leftarrow (-a1/norm);\ b[i] \leftarrow (-b1/norm);\}$
else $\{a[i] \leftarrow (a1/norm);\ b[i] \leftarrow (b1/norm);\}$
}

$$(A, B) \leftarrow \left(\sum_{i=1}^{k} a[i]/count, \sum_{i=1}^{k} b[i]/count,\right)$$

$linearity \leftarrow \sqrt{A^2 + B^2}$;
Repeat entire **for** loop for $angle1 \leftarrow angle + \pi/2$, which results in a new linearity value, *linearity1*;
If $(linearity < linearity1)$ **then** $linearity \leftarrow linearity1$;

The linearity measure $\sqrt{A^2 + B^2}$ produces numbers in the interval $[2/\pi, 1]$ (for a circle it is $2/\pi \approx 0.636$), as proven in Appendix I. This is normalized to [0, 1]:

$Linearity \leftarrow (linearity - 2/\pi)/(1 - 2/\pi)$;
Output *linearity*.

3.2. Rotation correlation/eccentricity

Correlation is a standard tool in statistics for determining whether there is a relation between two sets of points. If we consider the x and y values of points in a space separately, and apply correlation, we can directly measure linearity. Again, we first find the center of mass of the set of points along with its orientation. In this algorithm, the curve in question is rotated so that its new orientation is at an angle of 45° from the x-axis. Correlation is then done on the rotated curve. The linearity measure is the absolute value of the measured correlation of points (x_i, y_i) on the rotated curve.

Algorithm. *Rotation correlation.*

Input: array of points: $Points = (X_i, Y_i),\ 1 \leqslant i \leqslant count, k$;
Output: *linearity* value;

Find center of mass (X_c, Y_c) using moments;
Find angle of orientation *(angle)* by the above formula;
$angle1 \leftarrow angle + \pi/2$;
$anglerot \leftarrow \pi/4 - angle$; //*anglerot is angle of rotation of set of points to make them* 45° *to x axis.*

Rotate all points in the *Points* array by (*anglerot*) with respect to the origin, call new array *rotPoints*;
//Points in array *rotPoints* will be referenced as (x'_i, y'_i).
Find Correlation of points $(x'_i, y'_i),\ 1 \leqslant i \leqslant count$;

If $(Correlation < 0)$ **then** $Correlation \leftarrow -Correlation$;

Repeat entire procedure for *angle1*;
In this case, $anglerot \leftarrow \pi/4 - angle1$;
Find correlation (*Correlation1*) according to new *anglerot*;
If (*Correlation*>*Correlation1*) **then** *Linearity*←*Correlation*;
Else *Linearity* ← *Correlation1*;
Output *linearity*; //already normalized to [0, 1]

Eccentricity was the simplest measure to adapt to linearity that we could find. It was also used in Ref. [9]. The output of this algorithm is already in the interval [0, 1], so there was no need to normalize it. For a disc, this measure outputs 0, for a line, it outputs 1 since lines are eccentric.

Algorithm. *Eccentricity.*

Input: array of points: $Points = (X_i, Y_i),\ 1 \leqslant i \leqslant count$;
Output: Linearity value;

Find center of mass (X_c, Y_c) using moments;
Find second order moments: μ_{11}, μ_{02}, and μ_{20};

$$linearity \leftarrow \frac{\sqrt{(\mu_{20} - \mu_{02})^2 + 4\mu_{11}^2}}{\mu_{20} + \mu_{02}}.$$

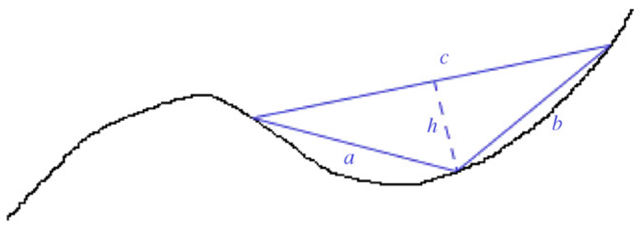

Fig. 2. Triangle formed by 3 random points, and its height h.

We will now prove the following theorem:

Theorem 1. *Rotation correlation and eccentricity always yield the same linearity measures.*

Proof. It is well known that the correlation measure is invariant to translation of data. Therefore we can translate data so that center of mass is moved to the origin, and thus $\mu_{01} = \mu_{10} = 0$. The correlation measure is then transformed into the following form:

$Correlation0 = \mu_{11}/\sqrt{\mu_{20}\mu_{02}}$.

When the orientation line coincides with the x-axis, the angle is 0, and from $0.5 * \arctan(2\mu_{11}/(\mu_{20} - \mu_{02})) = 0$ we obtain $\mu_{11} = 0$. Rotation around the origin for an angle A moves a point with coordinates (x, y) to point $(x \cos A - y \sin A, x \sin A + y \cos A)$. We have applied rotation for the angle $\pi/4$. When this is applied to every point from the set, the correlation will change to $correlation1 = (\mu_{20} - \mu_{02})/(\mu_{20} + \mu_{02})$, which can be verified by straightforward algebraic manipulation.

On the other hand, the linearity by eccentricity formula is invariant with respect to rotation. Consider the case when the orientation line coincides with the x-axis. Then $\mu_{11} = 0$ and the formula is transformed to $|\mu_{20} - \mu_{02}|/(\mu_{20} + \mu_{02})$. This is the same formula as $|correlation1|$. The possible change in sign has been corrected by the algorithm that only considers the absolute value of correlation, and therefore the two methods always give the same result. □

3.3. Triangle heights

Here, we take k triplets of random points from the set and compute the heights h to the longest side of the triangles that the triplets form. This h value is divided by the longest side c of the triangle to normalize the measure. This value is called hc. We use the average of these khc values as a linearity measure of the set of points. Fig. 2 illustrates this point.

Obviously, a low average of hc would represent a linear set of lines. Therefore, the average hc value is adjusted to fit the norm of higher linearity values representing linear sets of points. The minimum value of hc is 0. The maximum ratio for a height of a triangle is obtained in an equilateral triangle. In such cases, $h = \sqrt{3}a/2$, where a is the length of a side of an equilateral triangle, and the ratio is $\sqrt{3}/2$. To define a measure that will allocate 1 to linear points, and 0 to the considered case of three vertices of an equilateral triangle, each hc value is adjusted as follows:

$hc = 1 - (2hc/\sqrt{3})$.

The range of obtained linearity values of this algorithm are still in the range of (0.66, 1) for the examples that we tested. The minimum value of 0.66 is obtained for circles. We stretch out this interval by adjusting the linearity as follows:

$linearity = (hc - 0.66)/0.34$.

This adjustment produces results in the range of [0, 1].

Algorithm. *Triangle heights.*

Input: array of points: $Points = (X_i, Y_i)$, $1 \leqslant i \leqslant count, k$;
Output: *linearity* value;

Int *count*; *//number of input points on the curve*
Float *sumh* ← 0; *//sum of all h values*

For 1 **to** k **do** {//k is set to 500
  Take (x_1, y_1), (x_2, y_2), (x_3, y_3) from *Points* at random;
  Find distances between them: a, b, c; $//a \leqslant b \leqslant c$
  Find the equation of line passing through the two selected points with distance c, in the form $Ax + By + e = 0$;
  //Let point (u, v) be the point from the triplet (x_1, y_1), (x_2, y_2), (x_3, y_3), that is not on the line c.
  Height $h \leftarrow (Au + Bv + e)/\sqrt{A^2 + B^2}$.
  If $(h < 0)$ **then** $h \leftarrow -h$;
  $h \leftarrow h/c$;
  $h \leftarrow 1 - (2h/\sqrt{3})$
  $sumh \leftarrow sumh + h$;
}
$linearity \leftarrow sumh/k$;
$linearity \leftarrow (linearity - 0.66)/0.34$;
if $linearity < 0$ **then** $linearity \leftarrow 0$.

3.4. Triangle perimeters

This method is similar to the previous one in the sense that we take k triplets of random points from the set and compute a variation of the perimeters of the triangles that the triplets form. The three sides of the triangle are labeled a, b and c, where $a \leqslant b \leqslant c$. The measure that we are interested in is $p = (2c - a - b)/c$. If these three points form a triangle which is degenerate in the form of a line, then p tends to 1.

The minimum value is 0 for the vertices of an equilateral triangle. We take the average value p to measure linearity. The linearity measure of circles is found to be 0.76. The value 0.76 was the lowest obtained p value in our experiments, and was therefore mapped to 1. Therefore, we need to stretch the resulting numbers over the interval [0, 1]:

$linearity = (p - 0.76)/0.24.$

Although p can have values of 1 in theory (for the set of 3 points which are the vertices of an equilateral triangle), in practice the random triplets rarely produce an equilateral triangle, and experimentation shows that it is best to stretch the linearity interval using the parameters shown.

Algorithm. *Triangle perimeters.*

Input: array of points: $Points = (X_i, Y_i),\ 1 \leqslant i \leqslant count, k$;
Output: Linearity value;

Float $sump \leftarrow 0$; //sum of all p values
For 1 **to** k **do** { //k *is set to* 500
Take (x_1, y_1), (x_2, y_2), (x_3, y_3) from *Points* at random;
Find distances between them: a, b, c; //$a \leqslant b \leqslant c$
$p \leftarrow (2c - a - b)/c$.
$sump \leftarrow sump + p$;
}
$linearity \leftarrow sump/k$;
$linearity \leftarrow (linearity - 0.76)/0.24$
if $linearity < 0$ **then** $linearity \leftarrow 0$.

3.5. Contour smoothness

The original *smoothness* formula in Ref. [9] was defined as $4\pi S/P^2$. In their formula, S is the area of the shape, and P is its perimeter. This is another measuring scheme that was adapted for linearity. It bases its measurements on the area of a shape divided by the square of its perimeter. This measure was inspired by the compactness measure. We did not take the area of the entire shape into consideration at once. Instead, we once again applied our technique of sampling the point set by taking triplets of points, and averaging out their triangular areas. Each triplet of points produces a *smoothness* value in the form of $area/perimeter^2$. The maximum value for area divided by the triangle perimeter is $\sqrt{3}/36$ (for an equilateral triangle). After *smoothness* values are averaged to produce value *sums*, the result is adjusted as follows:

$sums = 36\ sums/\sqrt{3}.$

This limits *sums* to a value of 1. We reversed the meaning of this *smoothness* measurement by taking the compliment of the obtained value. The measured value for circles then became 0.45. The final measure is obtained by stretching out the result on the interval [0, 1], where

$linearity = (1 - sums - 0.45)/0.55.$

Algorithm. *Contour smoothness.*

Input: array of points: $Points = (X_i, Y_i),\ 1 \leqslant i \leqslant count,\ k$;
Output: Linearity value;

Float $sums \leftarrow 0$; //sum of all s values
For 1 **to** k **do** { //k is set to 500
Take 3 points (x_1, y_1), (x_2, y_2), (x_3, y_3) from *Points* array; //points should not be close to each other.
Find distances between them: a, b, c; //$a \leqslant b \leqslant c$

$p \leftarrow a + b + c$;

Find area of the triangle formed by these 3 points: *area*;
Float $s \leftarrow area/(p^* p)$;
$sums \leftarrow sums + s$;
}
$sums \leftarrow (sums^* 36)/\sqrt{3}$; //normalization factor
$linearity \leftarrow (1 - sums - 0.45)/0.55$;
if $linearity < 0$ **then** $linearity \leftarrow 0$.

3.6. Ellipse axis ratio

Here, we use the idea of measuring rectangularity as proposed in Ref. [6], and adapt it to measuring linearity. The concept is similar to the eccentricity measurement. We first find the center of mass and the first and second order moments of the set of input points, and then find the values of the major and minor axis of the best fit ellipse as determined by the formulas in Ref. [6]. The linearity value is given as 1-*minor axis/major axis*.

Algorithm. *Ellipse axis ratio.*

Input: array of points: $Points = (X_i, Y_i),\ 1 \leqslant i \leqslant count,\ k$;
Output: Linearity value;

Find center of mass (X_c, Y_c) using moments;
Find moments: μ_{00}, μ_{11}, μ_{02}, *and* μ_{20};
Find value for major axis a; //see literature review
Find value for minor axis b; //see literature review
$linearity \leftarrow 1 - b/a$.

4. Experimental data

We develop seven algorithms which assign linearity measures to finite sets of points (two of them are identified to be the same). These algorithms are implemented on Windows machines in C + + using Intel's computer vision library of basic functions called OpenCV. The input to each algorithm is a black and white image of size 400 × 400 pixels with white pixels representing the background, and black pixels representing the curves (set of pixels) to be tested for linearity. Each point in the image can be referenced with two integers (x_i, y_i).

The set of test images is seen in Fig. 3. All 30 are examined by each algorithm. Their linearity values are presented in Table 1. All of the digital points in the solid figures (circle, hexagon, etc.) are taken into consideration when evaluating linearity. Linearity values for "solid" shapes are very similar to the linearity values of just their borders.

The basic framework of each algorithm involved extracting the black pixels from the image and putting them into an

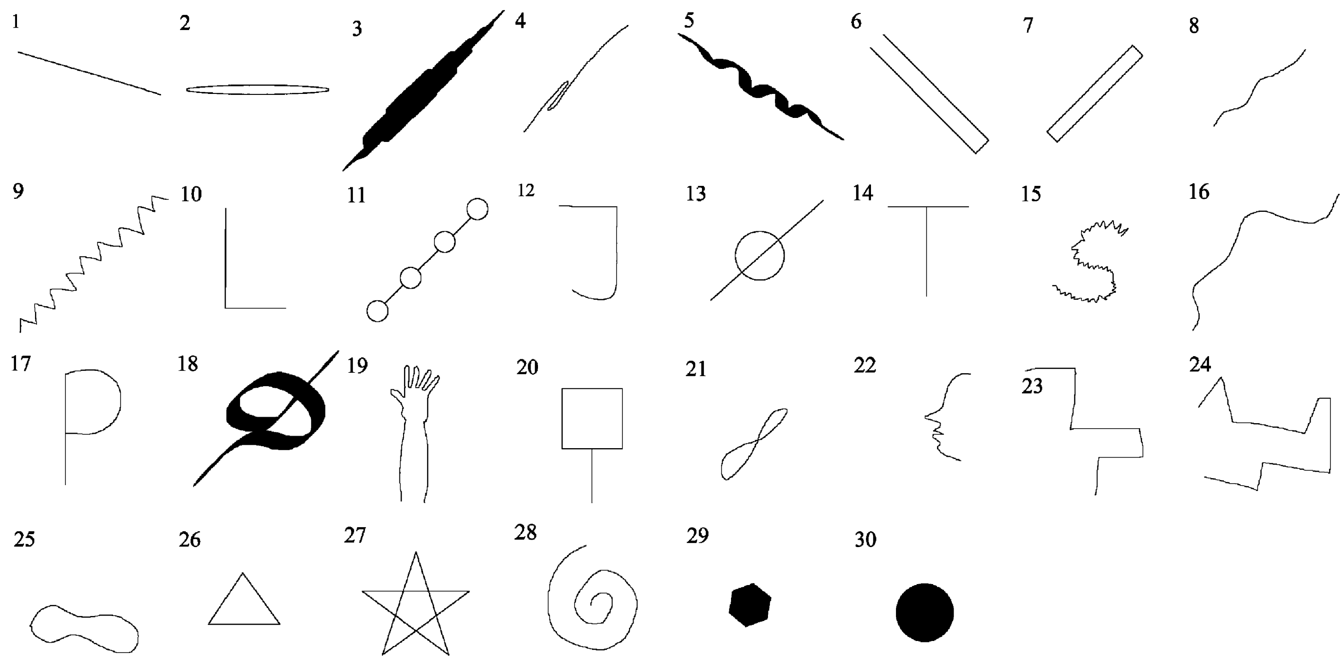

Fig. 3. Test examples.

Table 1
Results of linearity algorithms

	AO	RC/E	TH	TP	CS	EAR	AHP	SDH	SDA
1	99	100	99	100	98	100	100	0.0	0.8
2	88	99	75	86	69	90	86	16.8	10.8
3	83	97	64	78	57	87	83	17.0	14.9
4	91	99	82	91	77	94	81	16.3	8.1
5	88	98	74	86	67	91	81	11.8	11.4
6	73	92	53	64	45	80	78	17.9	17.4
7	80	93	59	71	51	81	78	18.3	15.5
8	95	99	85	96	79	93	76	8.9	7.6
9	85	98	75	88	68	90	71	18.9	10.8
10	59	73	58	69	51	61	66	20.9	8.0
11	81	97	70	82	63	83	62	23.7	11.7
12	42	65	42	53	33	54	61	21.2	11.4
13	61	83	40	49	35	70	56	21.1	18.4
14	27	50	13	17	10	43	54	22.6	16.5
15	72	86	64	81	53	72	52	22.8	11.8
16	53	61	7	41	32	12	50	21.7	21.7
17	49	71	30	39	24	59	49	25.1	17.8
18	55	83	36	46	29	69	45	17.8	20.4
19	29	48	19	25	16	41	40	16.5	12.5
20	74	92	53	65	46	79	40	20.3	17.0
21	62	79	55	71	44	66	35	25.2	12.3
22	49	71	33	43	26	59	27	18.7	16.6
23	17	28	15	19	12	25	22	17.1	6.1
24	56	74	26	36	20	61	20	16.4	21.4
25	12	19	10	6	13	17	20	18.1	4.7
26	1	1	2	3	2	1	18	18.3	0.8
27	44	79	12	57	42	65	17	13.4	23.1
28	7	14	1	4	0	13	15	19.5	6.0
29	2	6	10	13	7	6	8	7.3	3.8
30	3	0	6	1	3	0	0	0.9	2.3
Corr:	0.859	0.803	0.856	0.833	0.849	0.809			

unordered list. This list of unordered points would be passed to each algorithm for processing. The output of each algorithm would be a real number in the interval [0, 1] identifying the linearity measure of the set of points. A 1 means perfectly linear, whereas lesser values mean the shapes are less and less linear.

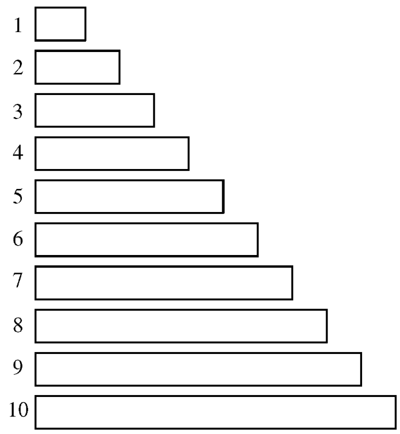

Fig. 4. Convergence test lines.

The table outlining the results of all six algorithms is shown below. The columns are labeled after the algorithms: average orientations (AO), rotation correlation and eccentricity (RC/E), triangle heights (TH), triangle perimeters (TP), contour smoothness (CS) and ellipse axis ratio (EAR). The average human perception column (AHP) shows the average results per figure of human measurements. The standard deviation of human perception (SDH) is seen in the next column, followed by the standard deviation of algorithms (SDA).

The comparison to human perception was done by correlating the results of each algorithm with the AHP column. The correlation values of each algorithm to the average human measurements are seen at the bottom of the table. According to these measurements, we conclude that the AO algorithm produced the best results. All of the algorithms produced relatively similar results, but the RC/E method showed itself to be the weakest when compared to the human average. The k value for the AO, TH and TP algorithms was 500 for the results seen in Table 1.

4.1. Convergence of linearity measures to 1

We also tested all of the linearity measures on a set of images seen in Fig. 4. The images in Fig. 4 are all rectangles that become progressively elongated. The first rectangle is 30 pixels wide and 20 pixels high. Each successive rectangle has the same height as the first one, but is 20 pixels wider than its predecessor. These test images are devised such that each image is more linear than its predecessor and less linear than its successor.

Linearity algorithms that function well should all produce linearity measures that follow this trend. The results of the linearity algorithm on the above given test set are seen in Table 2. We notice that all of the algorithms generally follow the trend of increasing linearity values. There are, however, some discrepancies in the results in the algorithms that rely on sample size such as AO, TH and TP. The RC algorithm shows continuously growing linearity measures since it is deterministic and does not rely on sample set size. Overall, all of the algorithms converge to perfect linearity.

Table 2
Convergence results

	AO	RC	TH	TP	CS	EAR
1	52.9	68.5	55.2	66.5	47.5	56.8
2	60.9	86.9	70.7	78.6	64.7	73.6
3	77.4	93.7	80.6	88.2	75.8	82.0
4	87.6	96.5	85.7	91.2	82.0	86.7
5	87.5	97.9	86.6	91.8	83.1	89.6
6	86.1	98.6	90.1	93.1	87.6	91.5
7	92.7	99.0	93.7	96.8	91.7	93.0
8	92.6	99.3	95.2	97.1	94.0	94.0
9	95.3	99.5	94.6	97.6	93.0	94.8
10	96.9	99.6	96.8	98.5	95.7	95.5

5. Conclusions

We proposed and tested several measures of linearity of finite point sets. This appears to be the first study of linearity in literature. In addition to computer vision applications, the proposed linearity measures have potential applications in manufacturing, for estimating the linearity of an axis of an object [10,11].

There are a number of possible extensions of this work. We believe that most of the presented measures can be extended to three dimensions and even further to arbitrary dimensions, to measure flatness of a finite set of points. We are currently extending this work to measure linearity of an ordered set of points. The proposed measures are adopted by also considering the ordering of points when projected along the orientation line. We are also applying such a measure to polygonization of curves.

Appendix A. Minimum average orientation measure

In Lemma 1, the number k of point pairs is approaching infinity, and the set of points on the circle is assumed to be infinite and consisting of all points on the circle perimeter. The average orientation measure in the following lemma and theorem refers to measurement before applying normalization, which brings the linearity measure of a circle from $2/\pi$ to 0.

Lemma 1. *Average orientation measure of set of points on the perimeter of a circle is* $2/\pi \approx 0.6366$.

Proof. For given orientation line with angle θ, and k pairs of points, with angles α_i, $1 \leqslant i \leqslant k$, the measure is

$$\sum_{i=1}^{k} |\cos(\alpha_i - \theta)|/k.$$

Because of symmetry, the measure remains the same for any orientation line. The measure is then

$$1/\pi \int_0^{\pi} \left(\sum_{i=1}^{k} |\cos(a_i - \theta)|/k \, \mathrm{d}\theta \right),$$

obtained when all possible orientation lines are considered. The later measure is equal to

$$1/(\pi k)\sum_{i=1}^{k}\int_0^{\pi}|\cos(\alpha_i-\theta)|/k\,\mathrm{d}\theta.$$

However,

$$\int_0^{\pi}|\cos(\alpha_i-\theta)|\,\mathrm{d}\theta=\int_0^{\pi}|\cos\theta|\,\mathrm{d}\theta$$
$$=\int_{-\pi/2}^{\pi/2}\cos\theta\,\mathrm{d}\theta$$
$$=2\int_0^{\pi/2}\cos\theta\,\mathrm{d}\theta=2.$$

Therefore the measure is

$$1/(\pi k)\sum_{i=1}^{k}2=2/\pi. \qquad \square$$

Theorem 2. *The average orientation measure of arbitrary object is* $\geqslant 2/\pi$, *with respect to at least one orientation line*.

Proof. The proof is by contradiction. Suppose that the linearity measure is $<2/\pi$ for all orientations. Let α_i, $1\leqslant i\leqslant k$, be measured sample orientations. Thus

$$\sum_{i=1}^{k}|\cos(\alpha_i-\theta)|/k<2/\pi$$

for any θ. Integrate this over all θ. Then

$$S=\int_0^{\pi}\sum_{i=1}^{k}|\cos(\alpha_i-\theta)|/k\,\mathrm{d}\theta<\pi 2/\pi=2.$$

Thus

$$S=\sum_{i=1}^{k}(1/k)\int_0^{\pi}|\cos(\alpha_i-\theta)|\,\mathrm{d}\theta<2.$$

Therefore, there exist i such that

$$\int_0^{\pi}|\cos(\alpha_i-\theta)|\,\mathrm{d}\theta<2.$$

But

$$\int_0^{\pi}|\cos(\alpha_i-\theta)|\,\mathrm{d}\theta=2$$

(see Lemma 1), which is a contradiction. $\square$

References

[1] M. Sonka, V. Hlavac, R. Boyle, Image Processing, Analysis, and Machine Vision, Chapman & Hall, London, 1993.

[2] J. Zunic, P.L. Rosin, A new convexity measure for polygons, IEEE Trans. Pattern Anal. Mach. Intell. 26 (7) (2004) 923–934.

[3] H.I. Stern, Polygonal entropy: a convexity measure, Pattern Recognition Lett. 10 (1989) 229–235.

[4] L. Boxer, Computing deviations from convexity in polygons, Pattern Recognition Lett. 14 (1993) 163–167.

[5] J. Zunic, P.L. Rosin, Rectilinearity measurements for polygons, IEEE Trans. Pattern Anal. Mach. Intell. 25 (9) (2003) 1193–1200.

[6] P. Rosin, Measuring rectangularity, Mach. Vision Appl. 11 (1999) 191–196.

[7] P. Rosin, Measuring shape: ellipticity, rectangularity, and triangularity, Mach. Vision Appl. 14 (2003) 172–184.

[8] P.L. Rosin, Measuring sigmoidality, in: Proceedings of CAIP 2003, Lecture Notes in Computer Science, vol. 2756, 2003, pp. 410–417.

[9] D. Csetverikov, Basic algorithms for digital image analysis, Course, Institute of Informatics, Eotvos Lorand University ⟨visual.ipan.sztaki.hu⟩.

[10] M.A. Badar, S. Raman, P.S. Pulat, Intelligent search-based selection of sample points for straightness and flatness estimation, J. Manufacturing Sci. Eng. 125 (2003) 263s–271s.

[11] E. Orady, S. Li, Y. Chen, Evaluation of minimum zone straightness by a nonlinear optimization method, J. Manufacturing Sci. Eng. 122 (2000) 795–797.

About the Author—MILOS STOJMENOVIC is currently working on his Ph.D. in Computer Science at the University of Ottawa. He completed his Master's thesis titled "Real time car detection with an AdaBoost machine learning approach and a small training set", at Carleton University, Ottawa in 2005. His research interests include Computer Vision, and wireless and ad hoc sensor networks.

About the Author—AMIYA NAYAK is an associate professor at School of Information Technology & Engineering, University of Ottawa since 2002. He received his B. Math. degree from University of Waterloo in 1981, and Ph.D. in computer engineering from Carleton University in 1991. He has over 15 years of industrial experience in software engineering in avionics and telecommunication applications, working at CMC Electronics and Nortel Networks. His current research interests are in the area of fault tolerance and testing of VLSI systems, distributed systems, and mobile ad hoc networks with over 70 publications in refereed journals and conference proceedings. He is in the Editorial Boards of International Journal of Parallel, Emergent and Distributed Systems and International Journal of Computing and Information Science.

About the Author—JOVISA ZUNIC received the MSc and PhD degrees in Mathematics and Computer Science from the University of Novi Sad (Serbia) in 1989 and 1991, respectively. He worked as a professor and researcher at the University of Novi Sad for more than a decade and is currently a senior lecturer in the Department of Computer Science at Exeter University. He has been a visiting researcher at several universities—most recently, in the Department of Computer Science at Cardiff University. He is also with the Mathematical Institute of the Serbian Academy of Sciences and Arts.
His research interest are in digital image analysis, digital geometry, shape representation and coding of digital objects, discrete mathematics, combinatorial optimization, wireless networks, and neural networks.