uOttawa

L'Université canadienne
Canada's university

FACULTÉ DES ÉTUDES SUPÉRIEURES ET POSTOCTORALES

uOttawa

L'Université canadienne
Canada's university

FACULTY OF GRADUATE AND POSDOCTORAL STUDIES

Milos Stojmenovic

AUTEUR DE LA THÈSE / AUTHOR OF THESIS

Ph.D. (Computer Science)

GRADE / DEGREE

School of Information Technology and Engineering

FACULTÉ, ÉCOLE, DÉPARTEMENT / FACULTY, SCHOOL, DEPARTMENT

Measuring conic properties and shape orientation of 2D point sets

TITRE DE LA THÈSE / TITLE OF THESIS

Amiya Nayak

DIRECTEUR (DIRECTRICE) DE LA THÈSE / THESIS SUPERVISOR

CO-DIRECTEUR (CO-DIRECTRICE) DE LA THÈSE / THESIS CO-SUPERVISOR

EXAMINATEURS (EXAMINATRICES) DE LA THÈSE / THESIS EXAMINERS

Miodrag Bolic

Nicola Santoro

Zoran Obradovic

Nejib Zaguia

Gary W. Slater

Le Doyen de la Faculté des études supérieures et postdoctorales / Dean of the Faculty of Graduate and Postdoctoral Studies

Measuring conic properties and shape orientations of 2D point sets

By:

Milos Stojmenovic

A Thesis submitted to
the Faculty of Graduate and Postdoctoral Studies
in partial fulfillment of
the requirements for the degree of
PhD in Computer Science

Ottawa-Carleton Institute for Computer Science
SITE
University of Ottawa
Ottawa, Ontario

May 2008

Your file Votre référence
ISBN: 978-0-494-46523-3
Our file Notre référence
ISBN: 978-0-494-46523-3

The undersigned hereby recommend to

the Faculty of Graduate and Postdoctoral Studies

acceptance of the Thesis

Measuring conic properties and shape orientations of 2D point sets

submitted by

Miloš Stojmenović

Dr. Iluju Kiringa
(OCICS, SITE)

Dr. Amiya Nayak
(Thesis Supervisor)

University of Ottawa,

May, 2008.

Table of Contents

Table of Figures

List of Tables

Abstract

We propose new methods for computing a shape's orientation and several shape measures for elongation, linearity, circularity, ellipticity, hyperbolicity, and parabolicity of 2D point sets. Measures for both ordered and unordered data sets which are invariant to rotation, scaling, and translation interest us. These measures should also be calculated very quickly. Moment based and average pair wise direction based calculations of orientation are proposed here. We describe linearity measures for unordered data sets called eccentricity, triangle perimeters, triangle heights, triplet smoothness, rotation correlation, average orientations, and ellipse axis ratio. Linearity measures for sorted data sets include average sorted orientations, triangle sides ratio, and the product of a new monotonicity measure and one of the existing measures for linearity of unordered point sets. The monotonicity measure is the ratio of signed and non-signed sums of piecewise projections onto the orientation line. In order to measure circularity, we transfer the Cartesian coordinates of the input set into polar coordinates. The linearity of the polar coordinate set corresponds to the circularity of the original input set given a suitable center. Our ellipse fit will determine the optimal location of the foci of the fitted ellipse along the orientation line (symmetrically with respect to the shape center) such that it minimizes the variance of sums of distances of points to the foci. In order to find ellipticity (hyperbolicity), we made use of the property that the sum (difference, respectively) of distances from each point on the ellipse to both foci is constant. We also propose an ellipticity measure based on the average ratio of distances of each point to the ellipse and to its center. The parabolicity measure is based on a similar idea of maintaining a constant sum of distances to the focus and a line parallel to the directrix line for each point. We discover that the definition of elongation highly correlates with the definition of linearity. All of the shape measures are tested on digital curves and compared with existing methods. All of the methods work in real time.

Acknowledgements

I would like to thank Amiya Nayak, Jovisa Zunic, Nishith Goel, Nicola Santoro, Nejib Zaguia, Nikola Bolic, Zoran Obradovic, OGS, NSERC, SITE, and my family and friends for having contributed, in one way or another, to the completion of this thesis.

Chapter 1 Overview

1.1 Overview of Shape analysis

Shape analysis has maintained constant interest in the research community for the last half century. It has however fostered even more enthusiasm lately since it has become a useful and practical component of larger computer vision and inspection systems. The real world application of shape analysis and recognition can be illustrated in the following examples. Recognition of rectangular and orthonormal line segment in satellite imagery corresponds to man made structures and settlements [ZR1]. Matching gradient features is used in handwritten character recognition [SS]. Detection of ellipses, circles, sine curves and other patterns is used in finding such things man made objects as cups, car wheels and car outlines [TCa]. Sets of points which appear linear 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. 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. 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.

1.2 Motivation

The main motivation for this work is in image processing. The aspiration here is to be able to identify, and orient shapes based on their point set representations in images. Measuring the characteristics of a finite set of points can become an interesting way of identifying the important components of a picture. The components or measurable properties we are referring to include linearity, circularity, ellipticity, hyperbolicity, parabolicity, and elongation.

The orientation of a shape is a property that interests us as well, but cannot be measured in the same way as the rest of the already mentioned shape properties. Each shape either does or does not have an orientation. Finding a shape's orientation is a common task in the area of computer vision and image processing, being used for example to define a local frame of reference and is helpful for recognition and registration, robot manipulation, etc. It is usually an initial step or a part of data pre-processing in many image processing and computer vision tasks.

In real world applications, the shapes we search for will not always be extracted as whole objects. We attempt to find algorithms that will be able to make sense of incomplete shape data. For example, the elongation measure for shapes whose boundary is not extracted completely should also be able to give useful results. Circularity is common in nature and industry, and finding a way of identifying it can be important for potential applications to both. Potential applications of this work are in computer vision systems that need to be able to identify circular characteristics of objects such as wheels of a car, or manufactured components, for more accurate detection, classification, and quality assurance. Like other measures for primitive geometric shapes, the measure of ellipticity and ellipse fitting are motivated by real world image processing problems. Ellipticity is common in nature and industry, and applications of ellipse identification are found in agricultural and medical imaging systems for identifying certain grains, onions, watermelons, cells, and even human

faces. Hyperbolicity and parabolicity are less common, but nonetheless have applications such as automatic industrial inspection and in x-ray diffraction imaging (where the centers of the atoms lie on hyperbolae).

Discovering more accurate and efficient algorithms for identifying relatively simple shapes and characteristics is an important step in improving the quality of every other application that makes use of shape detection. Shape detection and orientation can be viewed as the building blocks of almost any computer vision system. They have the potential to improve the efficiency and effectiveness of a variety of applications. Identifiable shapes may be used as features in machine learning systems for computer vision applications. Such computer vision systems are currently employed for tasks such as product quality control in manufacturing plants, face detection and identification in secure environments such as airports and casinos, and vehicle tracking/driving systems in robotics and traffic monitoring situations.

1.3 Problem Statements

While we are motivated by image processing applications, where the image is composed of pixels with integer coordinates, for simplicity we study the mentioned shape properties on point sets with real coordinates.

We propose several shape measures for computing a shape's orientation, and measuring its elongation, linearity, circularity, ellipticity, hyperbolicity, and parabolicity. We are interested in measures for both ordered and unordered data sets which are invariant to rotation, scaling, and translation of the input point set.

These measures should also be calculated very quickly. Wherever possible, each measure should output a shape result (such as linearity, or circularity) in the interval [0, 1]. The closer this result tends towards 1, the more that particular shape appears to have the measured characteristic.

Linearity measures how linear a point set is. A point set is perfectly linear if and only if all of the points in the set are collinear. Similar definitions apply in measuring each of the following: circularity, ellipticity, hyperbolicity and parabolicity. That is, a circularity value of 1 should be awarded to a shape if and only if there exists a circle so that all the points belong to that circle boundary.

A point set is ordered when all of the points are associated with their position in the list of all points, and these positions follow the shape, as seen in Figure 1.1. On the left of Figure 1.1, we see an elliptical shape as a collection of ordered points. On the right, we see the same shape as an unordered collection of points.

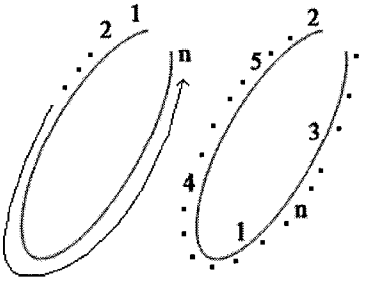


Figure 1.1 Ordered vs. Unordered point sets

The order of the points makes a difference when considering any shape measure. For example, in Figure 1.2 we see a shape as an ordered collection of points. If we were to measure the linearity of the shape in Figure 1.2 as an unordered collection of points, we would conclude that the shape is highly linear since it generally follows the straight blue line. However, if the shape was to be analyzed for linearity in an ordered point set sense, we would conclude that it is highly non linear since half of it follows the direction of the blue line, but the other half backtracks along the same line. Such a shape is non linear in terms of ordered linearity.

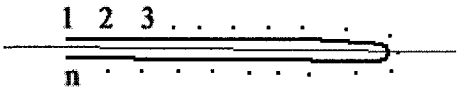


Figure 1.2 Highly linear vs. highly non linear

Orientation is a characteristic that defines the frame of reference of an object. The orientation line is the straight line that appears to be the best possible straight line approximation for a given point set. The orientation line normally passes through the center of gravity and is further specified by its angle. This angle often indicates the angle by which the shape needs to be rotated to view it in its natural position with a horizontal orientation.

Elongation will measure the relative amount that a shape has been stretched. This measure may not yield results in the interval [0, 1]; however, any measure given in an interval [a, b] may be adjusted to fit into the interval we choose. In the case of elongation, values e, where $e \in [1, \infty]$ are expected. When each resulting elongation value e is modified such that $e_1=1/e$, and possibly further by $e_2=1-e_1$, we obtain results in the desired interval [0, 1]. The concept of elongation is generally well understood. However, there is no clear, precise, mathematical definition of elongation in any dictionary we could find. We will study the link between the definition of elongation and the definition of linearity.

Fitting a conic section to a set of points (if possible) is an important step in finding a conicity measure for the set. We will study such fits of conic sections to evaluate their usefulness.

Conicity is a general ability of fitting and associating a set of points with its closest corresponding conic shape. Depending on the shape of the 2d point set, conicity may correspond to one of the following: linearity, circularity, ellipticity, hyperbolicity, or parabolicity, since they are all special cases of conics. Once the proper conic section fit is established, the evaluation of this fit is called its conicity. This is a new notion introduced in this thesis.

1.4 Assumptions and Limitations

Images are composed of pixels, which if processed correctly can result in interesting groupations of points that represent the interesting components in images. These interesting components, or groups of points, are worth identifying or at least classifying in the interests of being able to tell what is depicted in the original images. One of the assumptions we make deals with the type of measures we find in the image processing context. This means that we have to make choice between linearity and digital linearity for example. In our measurements, we deal with characterizing point sets with real coordinates. However, the applications that motivated this work have digital coordinates. In Figure 1.3, we see that the series of black pixels form a perfectly linear digital line (which digitizes the green line).

When we attempt to measure the linearity of the set of black pixels in Figure 1.3, we would not get a perfect linearity score since the pixels from this set are not collinear. Digital linearity, which is the study of whether pixels belong to a digital line, is a separate problem which is more complex, and is not examined here.

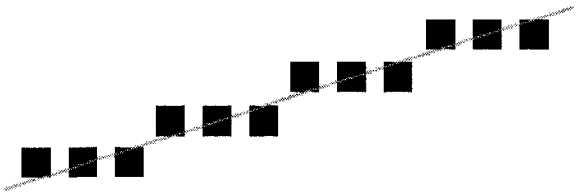


Figure 1.3 Linearty vs. Digital Linearity

Another assumption we choose is that all of our measures must be shape based as opposed to area based. This means that when finding shape orientation, or measuring any of the other properties of interest to us, the entire area of the shape (in pixels) is taken into account. We are looking for solutions that just consider the border, or perimeter of a shape when measuring or finding shape properties. Such approaches are referred to as being shape based. Figure 1.4 shows the input to a shape based algorithm vs. the input to an area based one. There are several advantages in pursuing a shape based approach. One of the main reasons is that shape based approaches are applicable to both open and closed shapes. They are also useful in measuring the shape characteristic of a disconnected set of points. Another point worth mentioning is that by simply considering border points, we deal with fewer pixels, and therefore perform any given task more quickly.

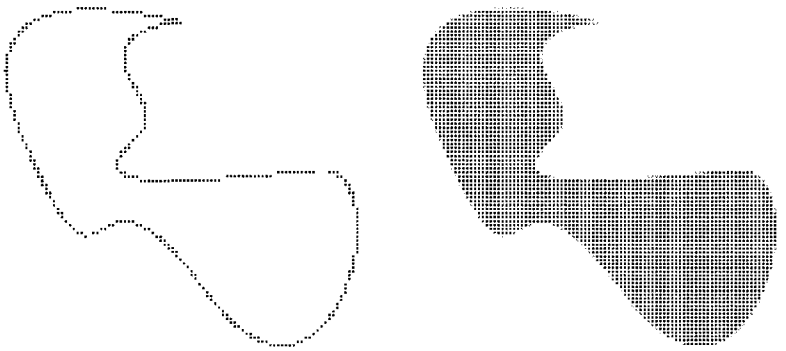


Figure 1.4 Shape based vs. area based algorithms

In our problem statement, we stated that all of our proposed algorithms must be invariant to rotation, scaling, and translation. For a measure to be invariant to rotation, the same result must be obtained for two sets of points, where one of the sets has been derived by rotating the other for an arbitrary center and angle of rotation. In order to be invariant to scaling, a measure must produce the same results for two sets S_1 and S_2 such that each point (x_1, y_1) in S_1 corresponds to point (fx_1, fy_1) in S_2, where f is a constant scaling factor. The same factor must be chosen in both directions since some affine transformations may deform the input set and influence the results of the shape measures. Being invariant to translation entails that the same result is obtained after a set of points has been translated to another location on the 2D grid. Since our motivation for this work lies in image processing, we will adapt this

restriction to be valid for the sets of points that are present in this discipline. When a discrete value point is rotated, scaled or translated, the result may be a real value point. However, in image processing, all points are discrete coordinates. Therefore, when a transformation is applied to a pixel, the result must be another pixel. This means that each transformation may yield an approximation of the real result. Nevertheless, since the images we deal with have a relatively high pixel resolution, this should not significantly affect the measures of point sets that have had transformations applied to them.

The shape measures may or may not be resistant to protrusions in the data set, depending on the application and the selected measure. Many measures rely on the idea that a shape's area can be compared to the area of its bounding box/disc in the hope of easily determining its nature (i.e. circular, rectangular). Such an idea performs poorly when shapes have protrusion, as seen in Figure 1.5. Here, we see a circular shape with a protrusion, and its circular bounding disc marked in light blue. The shape's area is much smaller that the area of its bounding disc, which results in a low circularity measure when the two are compared. In this example our measure should be resistant to protrusions.

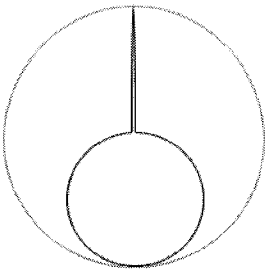


Figure 1.5 Circular shape with a protrusion

Although our primary goal is to recognize parts of a circle, ellipse, etc. as being perfectly circular (circularity value of 1), we will also consider measures that penalize shapes that have missing parts to accommodate for human perceptions. For instance, a set of collinear points may be considered as part of a circle with a very large radius. But most people would actually give such a shape a very low circularity score. This problem can be addressed by varying the perceived circle center. For instance, if the center of gravity is chosen as the shape center, the circularity measure of a line segment becomes zero.

We consider only points in the 2D (two dimensional) plane although some measures appear to be easily generalizable in 3D and even in higher dimensions.

1.5 Existing Solutions

Most of the existing solutions to the problems that we are studying are area based.

Orientation: Due to the variety of shapes as well as the diversity of applications there is probably no single method for computing shape orientation that could be efficiently and successfully applicable to all shapes. For that reason, several methods have been developed [CAT, HM, L2, L1, SI, SICT, TC]. Different techniques have been used, including those based on geometric moments, complex moments, and principal component analysis, for example. The suitability of those methods strongly depends on the particular situation to which they are applied, as they each have their relative strengths and weaknesses. The majority of the existing methods for computing the orientation are 'area based', i.e. the computation takes into account all points that belong to the shape not only the boundary

points. Among those area based methods, the most standard one says that shape orientation is determined by its axis of the least second moment of inertia [Ho, JKS, KR]. The axis of the least second moment of a shape is defined as a line that minimizes the integral of the squared distances of the shape points to the line. When working with shapes represented by a set of discrete points (set of pixels, for example) then the 'integral' should be replaced with the 'sum'. Also, because it is area based, the standard method is very robust with respect to noise and boundary defects.

Zunic [Z] defined the orientation of a polygonal shape P by the line that maximizes the total sum of squared values of the length of projections of the edges of the shape boundary onto this line. This method, called ZO here, can also be applied directly to open polygonal lines, or to a set of such polygonal lines. It has an advantage when working with shapes whose boundaries are not extracted completely, or with shapes with characteristic scribble details that have to be taken into account when a shape is being oriented. Since the method is boundary based it is not expected to be particularly robust, but the sensitivity of the method could be an advantage in some particular applications (e.g. when working with high precision inspection tasks).

The computed orientation is demonstrated to be in accordance with human perception in some standard cases. For example, a computed orientation of a rectangle is coincident with its longer edge. The computed orientation of shapes with one symmetry axis is coincident with such an axis or it is orthogonal to it, etc. The method is very simple to implement and moreover, it has been proven that the angle of such a defined shape orientation αZ can be computed from a simple formula

$$\tan(2 \cdot \alpha Z) = \frac{\sum_{i=1}^{n} |e_i|^2 \sin(2\alpha_i)}{\sum_{i=1}^{n} |e_i|^2 \cos(2\alpha_i)},$$

where P is a shape with a polygonal boundary, e_i, $(1 \le i \le n)$, are edges of the boundary of P and α_i, $(1 \le i \le n)$, are angles between the edges e_i and the x-axis.

Zunic, Kopanja and Fieldsend [ZKF] proposed three measures of boundary based shape orientability. They show how much orientation is sensitive to shape noise and digitization.

Elongation: In literature, shape orientation and shape elongation are strongly connected, and usually considered together [Ho, JKS, KR]. The standard measure of shape elongation is derived from the definition of shape orientation that is based on the axis of the least second moment of inertia. Precisely, the axis of the least second moment [Ho, JKS, KR] is the line which minimises the integral of the squares of distances of the points (belonging to the shape) to the line. However, since these methods deal in terms of moments of inertia, they are inherently area based. Let us mention that there are also some naive measures of elongation. For example, shape elongation can be measured as the ratio of the longer and shorter edges of the minimum area bounding rectangle for the measured shape. It is worth mentioning that such bounding rectangles are easy to compute [FS, MSa]. They however do not solve the problem since all bounding rectangle solutions are extremely sensitive to protrusions in the data set, which goes against our stated criteria.

Linearity: The closest research to measuring linearity deals with measuring rectangularity [R2].

Circularity: The contour smoothness measure was described in [C] as a measure of circularity, where the area of a shape is divided by the square of its perimeter. The most used circularity measure is the well-known $C=4\pi A/P^2$, where A is the area of the shape while P is its perimeter. It is scale independent. However, the border of an object may be highly irregular and the calculated value of C will then be very small and will not reflect the correct circularity measure, as observed in [DD].

Chatzis, Kaburlasos, and Theodorides [CKT] use the same C formula, but measure P differently. The perimeter is estimated using the number of eroded pixels. Each object is eroded two times using a 3x3 square structuring element first, and then a 3x3 cross-shaped structuring element to find the perimeter. The area is the number of pixels that belong to the object.

Lee and Sallee [LS] introduce a set of area based measures of circularity, triangularity, and rectangularity. Their circularity measure first separately calculates the intersection and union of the shape area S with the area of the circle C that best fits the shape. Their final circularity measure is the ratio of the areas of the intersection and union of S and C, $(S\cap C)/(S\cup C)$.

Proffitt [P] introduced measures for circularity and ellipticity on a digital grid. His circularity measure was based on taking the mean radius μ_r presumably from the center of gravity to each border pixel. The standard deviation σ_r of all such radii was also calculated. The proposed circularity measure was $P=\sqrt{1-(\sigma_r/\mu_r)^2}$. Haralick [H] proposed a similar earlier measure of σ_r/μ_r. This is the only shape based circularity measure that was found in literature.

Ellipticity: Ellipse fitting has been widely studied in literature. Voss and Süße [VS] described an area and moments based method of fitting geometric primitives. Rosin [R5, R6] discussed various ellipse fitting methods and various distance measures of points in data sets from the corresponding ellipse fit [R3]. We will discuss in detail a shape sampling based method [R6] by Rosin because his papers indicate it as the ultimate selection. Fitzgibbon, Pilu and Fisher [FPF] introduced the only existing direct fitting ellipse method which is heavily based on the work of Bookstein [B3].

Various ellipticity measures were proposed in literature. An ellipse fit is a prerequisite for some of them. These include DFT and the shape based measure by Proffit [P], the set operations area based method [KAK], and the Euclidean ellipticity shape based and orthogonal hyperbolae area based measures by Rosin [R3]. Other measures are not based on a prior ellipse fit, such as the elliptic variance shape based method [P], and the area and moment based method [R3].

Conicity: We will apply a shape sampling based method [R6] by Rosin because his papers indicate it as the best solution for conic fitting. The conic fit is done by taking n quintuplets of points as proposed by [R6]. Should these points be part of a conic shape, then each one should satisfy an equation of the form $ax^2+bxy+cy^2+dx+ey+f=0$, which is the standard equation of a conic in 2D space. This system is solved via Gaussian elimination to determine the values of the 5 coefficients. By repeating this procedure k times, and taking the median value for each coefficient, a more reliable fit is obtained. The coefficients themselves determine the type of conic obtained by the fit. The foci, center and orientation of the shape can be extracted from the 5 coefficients. Hyperbola fitting was studied in [B3, OZ], but they do not address the quality of the fit in these studies. To the best of our

knowledge, the measures of hyperbolicity, parabolicity and conicity are not proposed in literature.

1.6 Contributions

The contributions of this thesis can be summarized as follows.

1.6.1 Orientation

Several pure 'boundary based' methods, where the orientation of the shape is dependent on the boundary points are presented here. The first one uses the same formula based on moments, but is applied only on boundary pixels. It produces excellent results and is the method actually used here as one of the steps in some measures of linearity and circularity.

Another boundary based orientation method called Average_α ($A\alpha$) is introduced here. This method takes k pairs of random points (p_1, p_2) from the shape and finds the angles α that these lines p_1p_2 make with the x-axis. The orientation of the shape is the median α value.

In order to extend the class of shapes that can be oriented by the method ZO in [Z], we give a modification of the method. Instead of the squared length of the projections of the edges of the shape boundary we introduce a higher exponent. Precisely, we define the orientation of a polygonal shape by a line that maximises the total sum of the lengths of edge projections taken with the power of $2N$. In such a way, a suitable choice of N can preserve a computable orientation of a higher class of shapes. However, if the exponent $2N$ is fixed, there will always be shapes that cannot be oriented. Such shapes are M-fold (with $M > 2N$) rotationally symmetric shapes, for example.

Finally, we come to the shape orientation that satisfies an additional, very important, requirement. A modification was needed because of the requirement that increasing the density of sample points from the shape boundary should lead to the convergence of the computed orientations. The orientation of the polygonal shape is the line that maximizes the total sum of ratios of squared lengths of all the boundary edge projections onto this line over edge lengths. It has been proven that this modification was enough to preserve such a requirement. Such a new defined orientation has an additional benefit. The angle α of the orientation line can be derived from a closed formula via curvilinear integrals.

$$\tan(2\cdot\alpha)=\frac{\int_a^b \frac{2\dot{x}\dot{y}}{\sqrt{\dot{x}^2+\dot{y}^2}}dt}{\int_a^b \frac{\dot{x}^2-\dot{y}^2}{\sqrt{\dot{x}^2+\dot{y}^2}}dt}$$

that can be used for orientating shapes with curved boundaries given in a parametric form $\dot{x} = x(t)=dx/dt$, $\dot{y} = y(t)=dy/dt$, where x and y are functions of the parameter t, $t \in [a,b]$. The above formula can be applied for open curve arcs as well. This formula finds the orientation for shapes whose boundary is not extracted completely, which is impossible to achieve with existing area based computation. On the other hand, convergence is valid only for differentiable curves with real coordinates. For input sets with integer coordinates, convergence is invalid, and moreover the computed orientation might be far from a reasonable estimate. This is because the angles between neighboring pixels are 0^o, 45^o, 90^o or135^o, while the method heavily depends on projections and thus the accuracy of these angles.

1.6.2 Elongation

We define a new method for computing shape elongation for shapes with polygonal boundaries. The measure is the ratio of the maximal and minimal of the sums of squared lengths of the projections of all of the edges of the polygonal boundary onto a line which has a particular slope. We express the measure with a closed formula, whereas most existing methods for measuring any shape property are expressed via an algorithm. The below given formula expresses the new elongation measure. This measure finds the elongation for shapes whose boundary is not extracted completely, which is impossible to achieve with existing area based measures.

$$E(P)=\frac{\sum_{1\le i\le n}|e_i|^2+\sqrt{\left(\sum_{1\le i\le n}|e_i|^2\cdot\cos(2\alpha_i)\right)^2+\left(\sum_{1\le i\le n}|e_i|^2\cdot\sin(2\alpha_i)\right)^2}}{\sum_{1\le i\le n}|e_i|^2-\sqrt{\left(\sum_{1\le i\le n}|e_i|^2\cdot\cos(2\alpha_i)\right)^2+\left(\sum_{1\le i\le n}|e_i|^2\cdot\sin(2\alpha_i)\right)^2}},$$

where P is a shape with a polygonal boundary, e_i, $(1\le i\le n)$, are edges of the boundary of P and α_i, $(1\le i\le n)$, are angles between the edges e_i and the x-axis. Elongation values in the interval [0, 1] are computed as 1-1/E(P).

The new definition for measuring elongation of an arbitrary curve, is given below, and it involves first derivatives $\dot{x}=dx/dt$ and $\dot{y}=dy/dt$. Assume that we have a piecewise smooth enough curve p given in a parametric form $x = x(t)$, $y = y(t)$, ($t\in[a, b]$). The elongation $E(p)$ of the curve p is defined as

$$E(p)=\frac{Length(p)+\sqrt{\left(\int_a^b\frac{2\dot{x}\dot{y}}{\sqrt{\dot{x}^2+\dot{y}^2}}dt\right)^2+\left(\int_a^b\frac{\dot{x}^2-\dot{y}^2}{\sqrt{\dot{x}^2+\dot{y}^2}}dt\right)^2}}{Length(p)-\sqrt{\left(\int_a^b\frac{2\dot{x}\dot{y}}{\sqrt{\dot{x}^2+\dot{y}^2}}dt\right)^2+\left(\int_a^b\frac{\dot{x}^2-\dot{y}^2}{\sqrt{\dot{x}^2+\dot{y}^2}}dt\right)^2}}.$$

This computation is well defined when working with smooth curves described formally by the equations. On the other hand, the computation (estimating) of first derivatives when working with sample (discrete) data is usually a big problem. This remark should point out that an efficient estimation of $E(p)$ does not need computation (estimation) of the appearing derivatives. The elongation measure $E(P)$ can be estimated efficiently by a computation of $E(p) = E(A1,...Ak)$ if the set of sample points $P = \{A1, \ldots, Ak\}$ is dense enough on the curve p. The computation of $E(A1,...Ak)$ by using the equation for $E(P)$ is simple, fast, does not involve the first derivative approximations, and guaranties convergence $E(P)=E(A1,...Ak)\rightarrow E(p)$.

Generally speaking, all methods that use only boundary information must be sensitive to boundary changes (e.g. deformations, intrusions, noise, etc). Inevitably, once we accept to work with boundary information and exploit the benefits that come from sensitive methods, we have also to accept problems which could come from boundary sensitivity. Typical problems are caused by noise on images that should be processed. Some of the problems when dealing with a noisy shape can be avoided by a suitable choice of polygonal approximation or by applying some standard procedures (e.g. smoothing). Another

disadvantage of the method presented here could be the fact that the new elongation measure depends on the edge lengths and the edge orientations but not on the order of edges.

We would like to point out that the new measure is not developed in order to be dominant to the standard one. It is clear that when working with shape descriptors, it is not always possible to have a perfect or best measure. All shape descriptors have their strengths and their weakness while their usefulness is in a strong relation to the suitability of particular applications.

Additional elongation measures can be obtained if elongation is assumed to be the same property as unsorted linearity. This point of view was not found in literature, and is debatable.

1.6.3 Unordered Linearity

We propose 7 separate algorithms for testing linearity of unordered sets of points. They are called: average orientations, rotation correlation, triangle heights, triangle perimeters, triplet smoothness, eccentricity, and ellipse axis ratio. 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. 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. Triplet smoothness is based on the average ratio of area over squared perimeter of triangles formed by sample triplets of points from the set. Eccentricity is simple formula involving moments that was found in literature [C], and adapted here to finding linearity. We prove, theoretically and experimentally, that the eccentricity and rotation correlation methods give the same linearity measures. The ellipse axis ratio is based on the minor/major axis ratio of the best ellipse that fits the set of points.

1.6.4 Monotonicity

We propose a measure of monotonicity of ordered point sets. Monotonicity finds the orientation line of the set of points and measures the behaviour 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 in the set. For each ordered pair A_iB_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 $|A_iB_i|$ is compared against the sum of lengths $|A_iB_i|$.

1.6.5 Ordered Linearity

We put forward 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 the monotonicity measure multiplied by the corresponding linearity measures 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.

1.6.6 Circularity

We establish a new shape based algorithms that assign circularity values to both open and closed curves, and to both ordered and unordered sets of points. These measures are adaptations of the linearity measures. These linearity measures are applicable since the input set of points to our circularity algorithms was transformed from Cartesian representation to polar representation, where highly circular input point sets become highly linear in the new representation, for a proper choice of center. We first consider measures for unordered point sets. The circularity of ordered sets is obtained by multiplying the unordered results by their monotonicity factors. The choice of center of each shape influences its overall circularity value. The center of each shape is traditionally seen as its center of gravity. We also consider another definition of shape center here. The 'true center' (X_{tc}, Y_{tc}) of a shape is defined as the center of a circle C that best fits the shape to C. It is determined by sampling k triplets of points from the point set, and finding their true centers. The median center value of the k samples is taken as the shape's true center.

1.6.7 Ellipticity

We argue that the only existing direct ellipse fit method does not work properly and propose a new simple scheme. It will determine the optimal location of the foci of the fitted ellipse along the orientation line (symmetrically with respect to the shape center) such that it minimizes the variance of sums of distances of points to the foci. We next propose a novel way of measuring the accuracy of ellipse fits against the original point set. The evaluation of fits proceeds by our novel ellipticity measure which transforms the point data into polar representation where the radius is equal to the sum of distances from the point to both foci, and the polar angle is equal to the one the original point makes with the center relative to the x-axis. The linearity of the polar representation will correspond to the quality of the ellipse fit for the original data. We also propose an ellipticity measure based on the average ratio of distances to the ellipse and to its center. The choice of center for each shape impacts the overall ellipticity measure. We discuss two ways of determining the center of the shape. The measures are tested on a set of shapes. The proposed algorithms work well on both open and closed curves.

1.6.8 Parabolicity, hyperbolicity and conicity

This thesis describes the first known methods of measuring conicity, hyperbolicity and parabolicity of a set of points. After finding the best conic fit, we measure the corresponding ellipticity (using a known method), hyperbolicity or parabolicity value with respect to that best fit. We are interested in measures which rely exclusively on shape boundary points. They should also be calculated very quickly, be invariant to rotation, scaling and translation. The optimal fit and the location of the foci of the fitted shape is determined by taking n quintuplets of points and the medians of their coefficients as proposed in [R6]. The evaluation of fits transforms the point data into polar representation where the radius in this representation is equal to the sum of distances from each point to both foci (for ellipses), the difference of distances from each point to both foci (for hyperbolas), and the sum of

distances from each point to the focus and a line parallel to the directrix line (for parabolas). The polar angle is equal to the one the original point makes with the center of gravity of the shape in the case of ellipses and hyperbolas and the focus in the case of parabolas, relative to the x-axis. The radius and/or polar angle are then suitably transformed and normalized for the purposes of scalability. The linearity of the polar representation will correspond to the quality of the fit for the original data. The proposed algorithms work well on both open and closed curves, where applicable, and both connected and disconnected sets of points in the plane.

1.6.9 Linearity and elongation

We propose that the definition of elongation should overlap with the definition of linearity since we will show that these two measures produce results that are highly correlated when applied to different types of 2D shapes. Our experiments consist of testing known methods of linearity and elongation on sets of closed shapes contours, shapes whose areas are filled, and shapes with open contours. It was found that the Average Orientations linearity measure in this thesis best correlates to the elongation measures found in literature. It has a correlation value of above 0.9 with measures of elongation for open and closed curves. Also, we have discovered that the standard measure of elongation, applied to its intended area based shapes, gives almost identical results when it is applied to just the boundary pixels of the same area based shapes. They are over .98 correlated. This leads to a new linearity/elongation measure which is fast, applicable to both open and closed shapes, is given by a closed formula, and highly agrees with existing measures.

1.7 Performance Evaluation

Testing the performance of all of the algorithms presented here is a qualitative endeavour. When comparing two algorithms that are measuring the linearity of an irregular shape for example, how do we decide which one produced the more accurate result? In our case, we compare the new and existing measures to human measurements to see which algorithm was closest to them, but even that is not conclusive evidence of one algorithm being better than another. The most direct test of success for each algorithm is if it produces visually appealing results, which our algorithms did. We can conclusively say that our algorithms for testing linearity, hyperbolicity, parabolicity are the best that exist since they are the only ones of their kind in literature. The other point that separates our algorithms from the pack is that they are shape perimeter based measures. Elongation is the only shape based measure of its kind that exists in literature. The circularity and orientations algorithms are competitive with existing solutions.

Our algorithms were implemented and tested on sets of between 20 and 50 input shapes. Some of these shapes were open, and some were closed. The input to each algorithm is a black and white image of size 300x300 pixels with white pixels representing the background, and black pixels representing the curves (set of pixels) to be tested. Each point in the image can be referenced with two integers (x_i, y_i), and our test shapes had between 100 and 300 pixels each. The comparison of the measures to human perception was done by correlating the results of each algorithm with the average human perception. Almost all of the algorithms are implemented on Windows machines in C++ using Intel's computer vision library of basic functions called OpenCV. All of the algorithms are calculated very quickly (in milliseconds).

1.8 Publications resulting from the Thesis

1. Milos Stojmenovic, Amiya Nayak, Jovisa Zunic, Measuring linearity of a finite set of points, **IEEE International Conference on Cybernetics and Intelligent Systems (CIS)**, pp. 222-227, Bangkok, Thailand, 7-9 June 2006.
2. Milos Stojmenovic, Jovisa Zunic, New measure for shape elongation, IbPRIA 3rd Iberian Conference on Pattern Recognition and Image Analysis, **Lecture Notes in Computer Science,** Vol 4478, pp. 572-579, Girona, Spain, June 6-8, 2007.
3. Milos Stojmenovic, Amiya Nayak, Measuring linearity of ordered point sets, Pacific Rim Symposium on Image and Video Technology, **Lecture Notes in Computer Science**, Vol 4872, pp. 274-288, Santiago, Chile, 2007.
4. Milos Stojmenovic, Amiya Nayak, Shape based circularity measures of planar point sets, **IEEE International Conference on Signal Processing and Communications**, pp. 1279-1282, Dubai, UAE, 24-27 Nov 2007.
5. Milos Stojmenovic, Jovisa Zunic, Measuring elongation from shape boundary, **Journal of Mathematical Imaging and Vision**, Vol. 30, No. 1, pp. 73-85, 2008. (Journal version of paper 2)
6. Milos Stojmenovic, Amiya Nayak, Direct ellipse fitting and measuring based on shape boundaries, Pacific Rim Symposium on Image and Video Technology, **Lecture Notes in Computer Science**, Vol 4872, pp. 221-235, Santiago, Chile, 2007.
7. Jovisa Zunic, Milos Stojmenovic, Boundary based shape orientation, **Pattern Recognition**, Vol. 41, No. 5, pp. 1785-1798, 2008.
8. Milos Stojmenovic, Amiya Nayak, Conic properties of planar point sets, **International conference on Relations, Orders and Graphs: Interaction with computer science**, pp. 445-454, Mahdia, Tunisia, 2008 (Survey article).
9. Milos Stojmenovic, Amiya Nayak, Jovisa Zunic, Measuring linearity of planar point sets, **Pattern Recognition**, Vol 41, No. 8, pp. 2503-2511 (2008). (Journal version of paper 1)
10. Milos Stojmenovic, Amiya Nayak, Measuring the related properties of linearity and elongation of point sets, Iberoamerican Congress on Pattern Recognition, CIARP, Havana, Cuba, 9-12 Sept, **Lecture Notes in Computer Science** Vol. 5197, pp. 102-111, 2008.
11. Milos Stojmenovic, Amiya Nayak, Measuring conicity from shape boundaries, (to be submitted).

The candidate is the principal author for most of the material presented in this thesis, including literature reviews, new algorithms, new measures and experimentation. The assistance and direction of the supervisor, Dr. Amiya Nayak, and Prof. Jovisa Zunic from the University of Exeter (UK) is greatly appreciated.

Proofs of lemmas and theorems for this thesis are the contributions of the co-authors, and are left in the thesis for completeness.

Chapter 2 Literature review

We will describe several well known functions and notations on finite sets of points that are used in our shape measures here. These include the center of mass, moments and correlation. The rest of this section is dedicated to the existing work in orientation, elongation, and circularity. Other relevant shape measures such as rectangularity, eccentricity, sigmoidality, rectilinearity, and convexity are also reviewed since they are among the closest measures to those studies here.

2.1 Center of mass and moments

Moments and centers of mass are normally considered for closed shapes in R^2. The actual accurate values for the center of mass and moments are defined as follows.

$\overline{m}_{p,q}(S)$ are centralized moments of the shape S defined as

$$\overline{m}_{p,q}(S) = \iint_S (x - X_c)^p \cdot (y - Y_c)^q \, dxdy,$$

where (X_c, Y_c) is the center of mass in R^2 of shape S, calculated as

$$X_c = \frac{\iint_S x dxdy}{\iint_S dxdy}, \text{ and } Y_c = \frac{\iint_S y dxdy}{\iint_S dxdy}.$$

In image processing applications, these formulas are approximated by taking the set Q of pixels inside shape S. The center of mass of points in a set Q 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, and S is the number of points in Q.

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

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

where (x_c, y_c) is the center of mass of the set Q.

2.2 Orientation

Due to the variety of shapes as well as the diversity of applications there is probably no single method for computing shape orientation that could be efficiently and successfully applicable to all shapes. For that reason, several methods have been developed ([CAT, HM, L2, L1, SI, SICT, TC]). Different techniques have been used, including those based on geometric moments, complex moments, and principal component analysis, for example. The suitability of those methods strongly depends on the particular situation to which they are applied, as they each have their relative strengths and weaknesses. The majority of the existing methods for computing the orientation are however 'area based', i.e. the

computation takes into account all points that belong to the shape not only the boundary points. Among those area based methods, the most standard one says that shape orientation is determined by its axis of the least second moment of inertia ([Ho, JKS, KR]). The axis of the least second moment of a shape is defined as a line that minimizes the integral of the squared distances of the shape points to the line. When working with shapes represented by a set of discrete points (set of pixels, for example) then the 'integral' should be replaced with the 'sum'. Also, because it is area based, the standard method is very robust with respect to noise and boundary defects.

It is worth mentioning the practical value of orientability in human visual perception. For instance, orientable shapes can be matched more quickly than shapes with no distinct axis [Pa].

Precisely, the axis of the least second moment ([Ho, JKS, KR]) is the line which minimises the integral of the squares of distances of the points (belonging to the shape) to the line. The integral is

$$I(S,\varphi,\rho)=\iint_S r^2(x,y,\varphi,\rho)dxdy$$

where $r(x, y, \varphi, \rho)$ is the perpendicular distance from the point (x, y) to the line given in the form

$$x\cdot\cos\varphi - y\cdot\sin\varphi = \rho\,.$$

The angle α for which the above integral reaches a minimum defines the orientation of the shape S. It can be shown that such an angle α satisfies the following equation:

$$\tan(2\alpha)=\frac{\sin(2\alpha)}{\cos(2\alpha)}=\frac{2\cdot\overline{m}_{1,1}(\mathrm{S})}{\overline{m}_{2,0}(\mathrm{S})-\overline{m}_{0,2}(\mathrm{S})}=\frac{2\cdot\iint_S(x-X_c)\cdot(y-Y_c)dxdy}{\iint_S(x-X_c)^2dxdy-\iint_S(y-Y_c)^2dxdy},$$

where $\overline{m}_{p,q}(\mathrm{S})$ are centralized moments of the shape S. The angle α of orientation of the set of points Q is then determined by [C]:

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

where $\overline{m}_{i,j}(\mathrm{S})$ is normally approximated by μ_{ij}.

The problem is that there are many situations when this method does not give any answer as to what the shape orientation should be. There are many regular and irregular shapes where this standard method does not work ([TC, ZKF]). Also, while in many situations the robustness of a method is a desirable property sometimes it could be a disadvantage. It could happen that some (by the standard method) "nonorientable" shapes could be oriented by existing narrow intrusions into the shape interior or by small details on them. Those details correspond to a relatively small percentage of pixels (when working with digital images) and their impact is not easily detectable by robust methods. Note that rotation-symmetric and reflective-symmetric shapes appear very often not only in industry (as machine made products) but also in nature (e.g. human faces, crystals). The related problems (a detected symmetry axis could define shape orientation, for example) are intensively studied ([DG, L2, KK, L1, M, PY, SICT, TS, TC]), as well.

Zunic [Z] defined the orientation of a polygonal shape by the direction that maximizes the total sum of the squared lengths of the projections of all the shape edges onto a line defined by this direction – see Fig 2.1 for an illustration. The formal definition follows.

Definition 2.2.1 [Z] Let P be a shape with a polygonal boundary P. The orientation of P is defined by the angle α such that the total sum e is an edge of P

$$F(\alpha, P) = \sum_{e \text{ is an edge of } P} |pr_{\vec{a}}(e)|^2$$

of squared lengths of projections of the edges of P onto a line having the slope α is maximal.

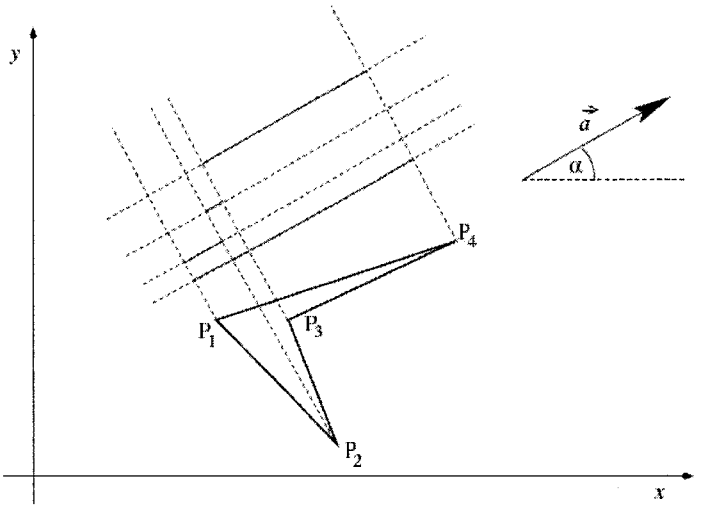


Figure 2.1 Projections of the edges of the polygon $P_1 P_2 P_3 P_4$ onto lines that have slope α

Theorem 2.2.1 [Z] Let P be an n-gon with edges e_i, $i = 1,\ldots, n$. Also, let α_i (for $i = 1, \ldots, n$) denote the angle between e_i and the x-axis. If the total sum

$$F(\alpha, P) = \sum_{i=1}^{n} |pr_{\vec{a}}(e_i)|^2$$

of the squared lengths of the projections of the edges e_i onto a line having slope α reaches its maximum then

$$\tan(2 \cdot \alpha) = \frac{\sum_{i=1}^{n} |e_i|^2 \sin(2\alpha_i)}{\sum_{i=1}^{n} |e_i|^2 \cos(2\alpha_i)}.$$

Since $F(\alpha, P)$ is a continuous function, it reaches its extreme values on the closed interval $[0, 2\pi]$. For a given polygon P those extreme values are easy to compute in accordance with the above formula. If the maximum is reached at point $\alpha = \alpha_0$ then the minimum is reached at the point $\alpha = \alpha_0 + \pi/2$.

Due to the simplicity of the method it is expected that there are situations when the method does not give an answer regarding what the shape orientation should be. Incidentally, this was already shown in the case of a regular triangle and in the case of a square. Zunic [Z] gives a formal characterisation of shapes for which the method does not

work. Let P be a shape with a polygonal boundary. We can conclude that $dF(\alpha, P)/d\alpha = 0$ *(i.e.* $F(\alpha, P)$ = constant) is equivalent to

$$\sum_{i=1}^{n} |e_i|^2 \cos(2\alpha_i) = 0 \text{ and } \sum_{i=1}^{n} |e_i|^2 \sin(2\alpha_i) = 0.$$

This is the "necessary and sufficient condition" for $F(\alpha, P)$ = *constant.* Consequently, if this holds, the method does not suggest any particular direction as a candidate for the orientation of P. Later on [Z] shows that $dF(\alpha, P)/d\alpha = 0$ holds for all shapes having more than two axes of symmetry or more generally, for all M-fold rotationally symmetric shapes, with $M > 2$.

Theorem 2.2.1 holds if P is an arbitrary polygonal curve (not necessarily a closed polygon). Moreover, the statement is valid if P consists of several not necessarily connected polygonal arcs. That is of importance when working with shapes whose orientation can be defined by small details. Non orientable shapes may become orientable just with the presence of noise or small inclusions in the data set.

2.3 Orientability

Zunic, Rosin and Kopanja [ZRK] discussed the degree to which a shape has a distinct (but not necessarily unique) orientation. Unlike the standard moment based measure of orientation, it is able to differentiate between the varying levels of orientability of n-fold rotationally symmetric shapes.

The problem becomes more complex when real shapes are replaced by their digitizations. Due to digitization, some non-orientable shapes may become orientable and vice versa. The computer orientation may depend strongly on the shape position with respect to the digitization grid and applied picture resolution. The computer orientation may be just a consequence of digitization or noise effects, or could be an inherent property of the considered shape.

Three orientability measures were proposed in [ZRK], each giving a number in the interval [0, 1]. A circle has the lowest orientability measure 0, and all measures tend to 1 when the shape approaches a straight line.

The first measure [ZRK] of orientability for a closed shape is:

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

Here, m_{ij} is an abbreviation for $\overline{m}_{i,j}(S)$. This formula gives results in 0 for all shapes that have more than two axes of symmetry. Their two other proposed measures do not have this disadvantage.

Let $R(\alpha)$ be the minimal enclosing rectangle for a given shape whose edges produce the angle α with the coordinate axes and which includes the given shape. Let A_{min} and A_{max} be the minima and maxima of $R(\alpha)$ for all angles α. The proposed measure of orientability [ZRK] is $1\text{-}A_{min}/A_{max}$. All shapes that have the same convex hull will have the same orientability measure. This can be modified if the following measure is considered [ZRK]:

$1-(A_{min} - \beta A)/(A_{max} - \beta A)$, where A is the area of given shape, and β is an arbitrary number between 0 and 1.

2.4 Elongation

In literature, shape orientation and shape elongation are strongly connected, and usually considered together ([Ho, JKS, KR]). The standard measure of shape elongation is derived from the definition of shape orientation that is based on the axis of the last second moment of inertia.

The minimum and maximum of $I(S, \varphi, \rho)$ are also easy to compute. They are:

$$\max_{\substack{\rho>0 \\ \varphi\in[0,2\pi]}} \{I(S,\varphi,\rho)\} = \frac{\overline{m}_{2,0}(S)+\overline{m}_{0,2}(S)+\sqrt{4\cdot(\overline{m}_{1,1}(S))^2+(\overline{m}_{2,0}(S)-\overline{m}_{0,2}(S))^2}}{2}$$

and

$$\min_{\substack{\rho>0 \\ \varphi\in[0,2\pi]}} \{I(S,\varphi,\rho)\} = \frac{\overline{m}_{2,0}(S)+\overline{m}_{0,2}(S)-\sqrt{4\cdot(\overline{m}_{1,1}(S))^2+(\overline{m}_{2,0}(S)-\overline{m}_{0,2}(S))^2}}{2}.$$

Let P be a shape with a polygonal boundary. Then, the elongation of P is defined as the ratio

$$ES(P) = \frac{\max\{F(\alpha,P) \mid \varphi\in[0,2\pi]\}}{\min\{F(\alpha,P) \mid \varphi\in[0,2\pi]\}}$$

of the maximum and minimum of the function $F(\alpha, P)$. It is the standard measure of elongation of the shape S. Some generalization of the standard method for measuring shape elongation can be found in [ZKF]. The standard measure (ES) of shape elongation is area based because all points belonging to the shape are involved in the computation (area moments are used).

Let us mention that there are also some naive measures of elongation. For example, shape elongation can be measured as the ratio of the longer and shorter edges of the minimum area bounding rectangle for the measured shape. It is worth mentioning that such bounding rectangles are easy to compute ([FS, MSa]). This measure is area based and is sensitive to protrusions.

2.5 Eccentricity

Eccentricity was described in [C], and it could also be used to measure the elongation of a shape. For a disc, this measure outputs 0, for a line, it outputs 1 since lines are eccentric. Eccentricity is defined as

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

2.6 Correlation

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

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

2.7 Measuring Rectangularity

One relevant 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) [R2]. 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 methods for measuring the rectangularity of regions are developed by Rosin [R2]. They are tested together with the standard minimum bounding rectangle 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:

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

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

2.8 Measuring Rectilinearity

Zunic and Rosin [ZR1] defined shape measures intended to describe the extent to which a closed polygon is rectilinear (each corner angle is 90^0 or 270^0). The two measures [ZR1] are based on the maximum ratio of perimeters measured by two metrics. One metric is the Euclidean distance while the other is city block distance (sum of differences in each coordinate). When 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 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.

2.9 Measuring Convexity

The most frequently used convexity measure in practice is the ratio between the area of polygon and area of its convex hull [SHB]. Zunic and Rosin [ZR2] discussed two measures that have advantages when measuring convexity of shapes with holes. [ZR2] first proposed

to measure the ratio of the largest convex polygon contained inside a given one, and the area of polygon, but noted that it is computationally expensive to apply. Then they proposed to measure the ratio of the Euclidean perimeters of a given shape its convex hull. Convexity measures were also studied in [S, B].

2.10 Measuring Sigmoidality

Rosin [R1] described several measures for sigmoidality. It is roughly a measure of 'S' shape where the 'fullness' (thickness) of the shape is not taken into account. Rosin [R1] 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. The second approach in [R1] is to consider (somewhat Gaussian) tangent angles. The function parameters are determined by matching mean absolute values and variances, area under curve is normalized to one, and correlation is used to measure sigmoidality. The third approach in [R1] 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 central line of symmetry.

2.11 Measuring the Collinearity of adjacent line segments

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 collinearity of four arbitrary points from the curve. It also does not give collinearity 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 to measure linearity.

2.12 Circularity Measures

The contour smoothness measure was described in [C] as a measure of circularity. It is defined as $C=4\pi A/P^2$, where A is the area of the shape while P is its perimeter. It is scale independent. However, the border of an object may be highly irregular and the calculated value of C will then be very small and will not reflect the correct circularity measure, as observed in [DD].

Di Ruberto and Dempster [DD] defined new circularity measures which are translation and scale invariant and are based on mathematical morphology. They [DD] make use of a distance function $dist(p)$ associated with each pixel p, as the integer valued radius of the smallest circle which erodes p. The regional maxima of the distance function represents inner points of f located at the longest distance from the border of f. Let $h=\max\{\text{dist}(p)\}$ be the maximum distance function, then V measure [DD] is defined as $V=\text{sum } dist(p)/h^3$, which is the ratio of the volume of the generated shape of the distance function and the cube of the height of the same function. T measure is defined as $T=A/h^2$ [DD], where A is the area of the object. The E measure in [DD] is $E=h/\text{sqrt}(A)$, where n is the radius of the largest disk contained inside the given object. V, T and E measures are sensitive to intrusions, since the intrusion affects h, the radius of the largest circle inside the shape. In the M measure in [DD], the distances from the center of gravity from the border in several directions (e.g. 8 or 16 cardinal directions) are calculated and the object deformity is calculated by the variance

of these distances. This measure is not invariant to rotations and can produce measure 1 for non-circular objects.

Proffitt [P] introduced measures for circularity and ellipticity on a digital grid. His circularity measure was based on taking the mean radius μ_r presumably from the center of gravity to each border pixel. The standard deviation σ_r of all such radii was also calculated. The proposed circularity measure was $P = \sqrt{1-(\sigma_r/\mu_r)^2}$. This is the only shape based measure that was found in literature.

Lee and Sallee [LS] introduce a set of area based measures of circularity, triangularity, and rectangularity. Their circularity measure first separately calculates the intersection and union of the shape area S with the area of the circle C that best fits the shape. Their final circularity measure is the ratio of the areas of the intersection and union of S and C, $(S \cap C)/(S \cup C)$.

Kim and Anderson [KA] defined a compactness measure (which is a different name for a circularity measure) for convex sets as the ratio of the area of a convex region and the area of the smallest circle containing the region. The measure is then modified and applied to measuring the compactness of images with respect to digital disks. Both convex sets and digital disks are outside the scope of this research. They [KA] also observed that the perimeter calculation for digital arcs, where neighboring edges have lengths 1 or $\sqrt{2}$, measure octagonality rather than circularity in images. A similar observation was made by Bottema [Bo] who then proposed a new digital circularity measure as the ratio $1-|A-D|/|A|$, where A is any discrete set and D is the discrete disk having the same area as A and the same center of mass. $A - D$ is the set difference of A and D, $(A \wedge \neg D)$, and $|X|$ is the number of pixels in X.

Chatzis, Kaburlasos, and Theodorides [CKT] described an image processing method to automate the particle size and shape measurement procedure. The method is based on fuzzy mathematical morphology. It is applied to estimate the size and the shape of fertilizers particles produced in a fertilizer industry. The particles tend to be spherical as much as possible. Image thresholding (threshold is minimal value of the smoothed histogram of the grayscale image) is applied to separate fertilized pixel from background. Connected components were then found, and circularity of each component is estimated as follows. The area is the number of pixels that belong to the object. The perimeter of each object is computed using a method based on fuzzy mathematical erosion. Each object is eroded two times using a 3x3 square structuring element first, and then a 3x3 cross-shaped structuring element. The perimeter is then estimated using the number of eroded pixels. The equal-area diameter of an object is defined as the diameter of a circle that has area equal to the area of the object. The circularity of an object is computed as the fraction between the equal-area diameter and another "equivalent diameter" computed as the diameter of an equal perimeter circle. The circle perimeter is calculated via the erosion based method. This circularity measure tends to the unit measure when the object tends to be a perfect circle. Note that this measure is in fact, similar to the square root of the traditional circularity measure. The difference is only that the perimeter is calculated differently.

The measures that satisfy our conditions of being resistant to protrusions, while at the same time being invariant to scaling, translation and rotation are: *C*, *P*, *LS* and *CKT*. *P* is the only one that is shape based. Note that CKT in fact measures digital circularity, but is added

here to approximate the circularity of ordered point sets when the points in the set are pixels with integer coordinates.

2.13 Elliptic and conic fitting

2.13.1 Conic fitting by sampling

Rosin [R6] described the least median of squares (LMedS) shape based approach to conic fitting. Several (k) minimal subsets (i.e. five points) of the data are selected at random and used to generate the ellipses through them. Should these points be part of a conic shape, then each one should satisfy an equation of the form $ax^2+bxy+cy^2+dx+ey+f=0$, which is the standard equation of a conic in 2D space. These five points each generate equations of the above form, which together form a block of five equations with 5 unknown. This system is solved via Gaussian elimination to determine the values of the 5 coefficients. Since we take k sets of quintuplets, we will have up to k valid solutions for the systems of equations which will produce up to k values for each of the 5 coefficients. The median values of each of a, b, c, d, e, and f of the solutions are chosen to represent the conic fit of the set of points. The coefficients themselves determine the type of conic obtained by the fit. This conic is an ellipse if $b^2-4ac<0$. It is a hyperbolae if $b^2-4ac>0$ and it is a parabola if $b^2-4ac=0$.

The center of the conic is $(X_c, Y_c) = ((cd - be)/(b^2 - ac), (ae-bd)/(b^2-ac))$. The angle of orientation of the shape is $\theta = (\arctan(b/(a-c)))/2$. The shape is then rotated to be parallel to the y-axis, which adjusts the coefficients as follows: $a' = a*\cos^2\theta + b*\cos\theta*\sin\theta + c*\sin^2\theta$; $b' = 0$; $c' = a*\sin^2\theta - b*\cos\theta*\sin\theta + c*\cos^2\theta$; $d' = d*\cos\theta + e*\sin\theta$; $e' = -d*\sin\theta + e*\cos\theta$; $f' = f$. From the rotated coefficients, we can calculate the values of the major and minor axes of the shape (if they exist). The squares of the major and minor axes: A^2, B^2 are calculated to be: $A^2 = d'^2/(4a'^2) + e'^2/(4a'c') - f'/a'$; and $B^2 = d'^2/(4a'c') + e'^2/(4c'^2) - f'/c'$. The minor axis of a hyperbola is an imaginary number but nonetheless is useful in identifying the value of C, which is the distance from the center to either focus: $C = \sqrt{|A^2 - B^2|}$. The points (X_c+C, Y_c), and (X_c-C, Y_c) specify the locations of the foci of the rotated shape. The true locations of the loci are obtained by rotating them back by the angle of orientation θ. The rotated, and true foci points of the original set of points are defined as: $F_1 (X_c + C*\cos\theta, Y_c + C*\sin\theta)$ and $F_2 (X_c - C*\cos\theta, Y_c - C*\sin\theta)$.

2.13.2 Moment based ellipse fitting

Voss and Süße [VS] described a moments and area based method of fitting geometric primitives by ellipses. The data is normalized into a unique canonical frame (a circle with specific radius for the case of ellipse fitting), by applying an affine transform. Given input points, they first find the centroid, and translate all points so that the new centroid is at the origin. Then $\mu_{01} = 0$ and $\mu_{10} = 0$ for the translated shape. They then apply an x-sheering transform which maps (x, y) to (x', y') as follows: x'=x+βy, y'=y. The goal is to obtain an elliptical shape with horizontal and vertical axis. For this position, the new moment value of m'_{11} is 0 because such an ellipse is an odd function. This means that

$$\iint_S (x+\beta y)y\,dx\,dy = 0, \text{ which decides } \beta = -\mu_{11}/\mu_{02},$$

where μ_{11} and μ_{02} are taken before the transform [RSV]. The elliptical shape should now be scaled to obtain a circular shape with new unit moments $M_{02} = M_{20} = 1$. The circle with such

unit value moments has radius $r = (4/\pi)^{1/4}$ [RSV]. The new transformation is $x''=\alpha x'$, $y''=\delta y'$.

$$1 = M_{20} = \iint_{S''} x''^2\, dx''dy'' = \iint_{S'} (\alpha x')^2\, \alpha dx' \delta dy' = \alpha^3 \delta \mu'_{20},$$

and similarly $1 = M_{02} = \alpha\delta^3 \mu'_{02}$. Moments μ'_{02} and μ'_{20} of the shape are calculated after x-shearing and before scaling. Multiplication gives $(\alpha\delta)^4 \mu'_{02}\,\mu'_{20}$, which leads to $\alpha = (\mu'_{02}/\mu'^3_{20})^{1/8}$, $\delta = (\mu'_{20}/\mu'^3_{02})^{1/8}$. This decides the new transform, which brings the data to a circular form with a unique canonical frame.

Applying an inverse transforms to the circle will result in getting the elliptical fit for the original data. A circle with radius r is scaled back to an ellipse with axes $a = r/\alpha$, $b = r/\delta$. The equation of that ellipse is $x'^2/a^2 + y'^2/b^2 = 1$. After inverse x-shearing, the equation of the original ellipse is $(x+\beta y)^2/a^2 + y^2/b^2 = 1$.

2.13.3 Direct least square fitting

Fitzgibbon, Pilu, Fisher [FPF] described a direct method for least square fitting of ellipses. The method is based on the method by Bookstein [B3] for fitting conic sections to scattered data. The method in [FPF] is ellipse-specific, so it always returns an ellipse.

The equation $\boldsymbol{a}'\boldsymbol{x} = ax^2+bxy+cy^2+dx+ey+f = 0$, where $\boldsymbol{a}=[a\ b\ c\ d\ e\ f]^T$ and $\boldsymbol{x} = [x^2\ xy\ y^2\ x\ y\ 1]^T$ describes an arbitrary conic section $\boldsymbol{a}$. This conic section is an ellipse if $b^2-4ac<0$.

Given a point (x_i, y_i), its algebraic distance to conic $\boldsymbol{a}$ from a point to a candidate ellipse is $\boldsymbol{a}\cdot\boldsymbol{x_i}$ where $\boldsymbol{x_i} = [x_i^2\ x_iy_i\ y_i^2\ x_i\ y_i\ 1]^T$. The method [B3] minimizes the sum of squared algebraic distances $G = \sum_{i=1}^{N} (\mathbf{a}\cdot\mathbf{x}_i)^2$, where $\boldsymbol{x_i}$ are the input points.

Let $\boldsymbol{D} = [\boldsymbol{x_1}\ \boldsymbol{x_2}\ \ldots\ \boldsymbol{x_n}]^T$, and $\boldsymbol{S}=\boldsymbol{D}^T\boldsymbol{D}$ (called the scatter matrix, a symmetric 6x6 matrix). Then $G = (\boldsymbol{Da})^T(\boldsymbol{Da}) = \boldsymbol{a}^T\boldsymbol{D}^T\boldsymbol{Da} = \boldsymbol{a}^T\boldsymbol{Sa}$. Therefore the problem is to minimize $G = \boldsymbol{a}^T\boldsymbol{Sa}$. However, a constraint on coefficients needs to be placed, since otherwise the solution is not unique. Bookstein [B3] suggested to use constraint $a^2 + b^2/2 + c^2 = 2$. We follow here the description and solution of [B3] since [FPF] only discussed the difference from that algorithm. This solution is generally applicable for any constraint of the form $\boldsymbol{a}^T\boldsymbol{Ca}$ = constant. The main contribution of [FPF] is to replace the diagonal constraint matrix $\boldsymbol{C}$ by a matrix which corresponds to the condition $4ac-b^2=1$, which then always produces an ellipse. Such a matrix $\boldsymbol{C}$ satisfies $\boldsymbol{a}^T\boldsymbol{Ca} = 1$ and has all zeros except $C_{13} = C_{31} = 2$, $C_{22} = -1$. Let $\boldsymbol{a}^T = (\boldsymbol{a_1}^T \mid \boldsymbol{a_2}^T) = [a\ b\ c \mid d\ e\ f]$. Let $\boldsymbol{S}$ be decomposed into four blocks, where $\boldsymbol{S_{11}}$, $\boldsymbol{S_{12}}=\boldsymbol{S_{21}}^T$ and $\boldsymbol{S_{22}}$ are 3x3 matrices. Then $G = \boldsymbol{a}^T\boldsymbol{Sa} = \boldsymbol{a_1}^T\boldsymbol{S_{11}}\boldsymbol{a_1} + 2\boldsymbol{a_1}^T\boldsymbol{S_{12}}\boldsymbol{a_2} + \boldsymbol{a_2}^T\boldsymbol{S_{22}}\boldsymbol{a_2}$. The constraint is on $\boldsymbol{a_1}$, not on $\boldsymbol{a_2}$. Assuming $\boldsymbol{a_1}$ is fixed, the minimum for G is obtained when $d(\boldsymbol{a}^T\boldsymbol{Sa})/(d\boldsymbol{a_2}) = 0$. Using theorems of matrix calculus, this means that $2\boldsymbol{a_1}^T\boldsymbol{S_{12}} + 2\boldsymbol{a_2}^T\boldsymbol{S_{22}} = 0$, or $\boldsymbol{a_2}^T = -\boldsymbol{a_1}^T\boldsymbol{S_{12}}\boldsymbol{S_{22}}^{-1}$. Then $G = \boldsymbol{a_1}^T\boldsymbol{S_{11}}\boldsymbol{a_1} + \boldsymbol{a_1}^T\boldsymbol{S_{12}}\boldsymbol{a_2} = \boldsymbol{a_1}^T(\boldsymbol{S_{11}} - \boldsymbol{S_{12}}\boldsymbol{S_{22}}^{-1}\boldsymbol{S_{21}})\boldsymbol{a_1} = \boldsymbol{a_1}^T\boldsymbol{S_1}\boldsymbol{a_1}$, $\boldsymbol{S_1} = \boldsymbol{S_{11}} - \boldsymbol{S_{12}}\boldsymbol{S_{22}}^{-1}\boldsymbol{S_{21}}$.

The problem now is to minimize $\boldsymbol{a_1}^T\boldsymbol{S_1}\boldsymbol{a_1}$ subject to $\boldsymbol{a_1}^T\boldsymbol{C'}\boldsymbol{a_1}=1$, where $\boldsymbol{C'}$ is the top 3x3 sub-matrix from $\boldsymbol{C}$ (nonzero elements are $C'_{13}=C'_{31}=2$, $C'_{22}=-1$). The Lagrange multiplier λ is introduced, to minimize $\boldsymbol{a_1}^T\boldsymbol{S_1}\boldsymbol{a_1} - \lambda\boldsymbol{a_1}^T\boldsymbol{C'}\boldsymbol{a_1}$. The derivative with respect to $\boldsymbol{a_1}^T$ implies $2\boldsymbol{S_1}\boldsymbol{a_1} - 2\lambda\boldsymbol{C'}\boldsymbol{a_1} = 0$. λ is a relative eigenvalue of $\boldsymbol{S_1}$ with respect to $\boldsymbol{C'}$, or a solution to $|\boldsymbol{S_1} - \lambda\boldsymbol{C'}|=0$, and $\boldsymbol{a_1}$ is the corresponding eigenvector. The determinant $|\boldsymbol{S_1} - \lambda\boldsymbol{C'}|=0$ is a cubic polynomial in λ. The eigenvectors for each real solution λ are obtained from $(\boldsymbol{S_1} - \lambda\boldsymbol{C'})\boldsymbol{a_1}=0$. Usually the best solution corresponds to the smallest λ [1]. Let $\boldsymbol{H}=\boldsymbol{S_1} - \lambda\boldsymbol{C'}$ and $\boldsymbol{u} = (a\ b\ c)^T$

be a solution of this homogeneous 3x3 system $\boldsymbol{Hu}=0$ (one can fix $c=1$). Then $\mu\boldsymbol{u}$ is also a solution for any μ, and $\boldsymbol{a_1} = \mu\boldsymbol{u}$ satisfies constraint $\boldsymbol{a_1}^T\boldsymbol{C'a_1}=1$. Thus $\mu^2\boldsymbol{u}^T\boldsymbol{C'u}=1$ and $\mu=(1/(\boldsymbol{u}^T\boldsymbol{C'u}))^{1/2}$. The conic we looked for is $\mu\boldsymbol{u} = (a\ b\ c)^T$ while $\boldsymbol{a_2}^T=(d\ e\ f)$ is obtained from $\boldsymbol{a_2}^T = -(\mu\boldsymbol{u})^T\boldsymbol{S_{12}}\boldsymbol{S_{22}}^{-1}$.

It can be observed, however, that scaling input data $\boldsymbol{D}$ leads to the scaling of the matrices $\boldsymbol{S}$ and $\boldsymbol{S_1}$ without changing constraint $\boldsymbol{C'}$ and therefore changes the cubic polynomial $|\boldsymbol{S_1} - \lambda\boldsymbol{C'}|=0$. This will change the claimed optimal values for λ and $\boldsymbol{a_1}$, as confirmed by our implementation, without ever producing a visually good fit. We believe that the optimal fit should be scalable and therefore conclude that the method [B3, FPF] does not work.

Note that [OZ] applied scaling of input data such that the average distance of a point from the shape center is $\sqrt{2}$. However, there is no mathematical support for the optimality of such scaling. Further, [FPF] argues that the method is very sensitive to noise, since all points impact the result.

2.14 Ellipticity measures

2.14.1 DFT based ellipticity measure

Proffitt [P] described a shape based approach for measuring ellipticity based on the discrete Fourier transform (DFT). An ellipse is fitted to the shape by centering it on the shape's centroid. The ellipse is then scaled such that its mean square of the lengths of the lines from the centroid to the boundary points matches the shape's. Let $u[j] = a[j] + ib[j]$, $i=\sqrt{-1}$, and $v[j] = x[j] + iy[j]$ be the corresponding points on the ellipse and the shape boundary, with the line between them passing through the centroid. Then the ellipticity measure D [P] (not normalized to [0, 1]) is $D^2 = \frac{1}{2N}\sum_{j=1}^{N}(u[j]-v[j])^2$. Note that $(u[j] - v[j])^2 = (a[j]-x[j])^2 + (b[j] - y[j])^2$ is the Euclidean distance.

2.14.2 Area comparison to measure ellipse fit

Koprnicky, Ahmed, and Kamel [KAK] defined ellipticity measures based on comparing the areas of shape S, the area of its ellipse fit R, and the areas of set differences $S\backslash R$ and $R\backslash S$ between the two. The measure that is closest to our criteria is $(\text{area}(S\backslash R)+\text{area}(R\backslash S))/(\text{area}(S \cup R))$. The authors do not elaborate on how to determine the best ellipse fit. This measure however might produce results that are outside of the interval [0, 1] when the ellipse fit is much bigger that the shape it is trying to fit. We therefore modified their method to measure ellipticity via $\text{area}(S \cap R) / \text{area}(S \cup R)$.

2.14.3 Moment based ellipticity measures

An area based ellipticity measure of a fit is obtained by comparing the differences between the central moments u_{ij} and u'_{ij} of the shape and the corresponding ellipse fit [R3, RZ]. The measure is

$$\frac{1}{1+\sum_{i,j=0}^{i+j\le 4}(\mu'_{ij}-\mu_{ij})^2}.$$

This method relies on normalizing u_{ij} coefficients, normally gives number close to 0, and can be used to rank shapes.

Rosin [R3] defined another moment based ellipticity measure as follows. Since any ellipse can be obtained by applying an affine transform to a circle, the simplest affine moment invariant of the circle can be used to characterize ellipses. It is defined as $I = (\mu_{20}\mu_{02}-\mu_{11}^2)/\mu_{00}^4$, where μ_{pq} are central moments. The moment for the unit radius circle is $I_c = 1/(16\pi^2)$. Thus I is measured for a given shape, and the measure of ellipticity of that shape in [R3] is:

$$E = 16\pi^2 I \text{ if } I \leq 1/(16\pi^2), \text{ or } 1/(16\pi^2 I) \text{ otherwise.}$$

2.14.4 Elliptic variance based ellipticity measure

Peura and Iivarinen [PI] described an 'elliptic variance' which they used to measure ellipticity based on shape perimeters. The center of gravity $\boldsymbol{u} = (u_1, u_2)^T$ and the covariance C of N data points $\boldsymbol{p}_i = (x_i, y_i)^T$ are calculated. Covariance is a 2x2 matrix and is calculated by the following formula: $\boldsymbol{C}=\frac{1}{N}\sum_{j=1}^{N}(\mathbf{p}_i-\mathbf{u})(\mathbf{p}_i-\mathbf{u})^T$.

The mean radius v of the contour is

$$v=\frac{1}{N}\sum_{i=1}^{N}\sqrt{(\mathbf{p}_i-\mathbf{u})^T\mathbf{C}^{-1}(\mathbf{p}_i-\mathbf{u})}.$$

The elliptic variance is then

$$EVAR=\frac{1}{Nv^2}\sum_{i=1}^{N}\left(\sqrt{(\mathbf{p}_i-\mathbf{u})^T\mathbf{C}^{-1}(\mathbf{p}_i-\mathbf{u})}-v\right)^2.$$

Rosin [R3] modified it to get a measure in [0, 1] as follows: $PI = 1/(1+EVAR)$. He observed that $EVAR$ suffers from the same problems as many distance approximations used for ellipse fitting. In particular, similar to standard algebraic distance, it exhibits a curvature bias (distances near the pointed ends of the ellipse are underestimated relative to distances at the flatter sections). This leads to irregularities at the ends having less of an effect than at the sides. However, it is not prone to the asymmetry between distances inside and outside the fitted ellipse.

2.14.5 Orthogonal hyperbolae distance and ellipticity measure

Rosin described a number of metrics for calculating the approximate error of each point, which represents the distance the point deviates from the fit [R5, R7]. The true point error – the distance from the point along the line normal to the ellipse, involves solving a quartic equation. Rosin recommended [R3] using the method that approximates the normal distance to an ellipse by the distance to the intersection with the confocal orthogonal hyperbolae passing through the point [R5]. The ellipse fit error is then robustly and accurately determined by the summed errors of the ellipse fit as

$$SE = \sum_{i=1}^{N} d_i ,$$

where d_i are the orthogonal hyperbolae based distance approximations [R5] and N is the number of points to fit. Rosin [10] then proposed the following area based ellipticity measure: $EE=(1+SE/(N\sqrt{A}))^{-1}$ (A is the area of given shape).

2.14.6 Detecting elliptical shapes

[TR] described a variety of shape measures, applying them as shape descriptors of saccular otoliths. They list a number of shape measures from literature for measuring aspect ratio (elongation), compactness, convexity, eccentricity, rectangularity, ellipticity, circularity [P], triangularity [VS], convexity, intrusiveness, protrusiveness, and several further measures.

[D, ZL] consider the problem of detecting ellipses, but do not measure ellipticity in the process. Davies [D] studied the problem of detecting cereal grains considering them as ellipses with aspect ratios (elongation) close to 2:1. They detect ellipses based on the property that the line passing through the centers of two horizontal chords also passes through the center of ellipse. A Hough transform was applied to handle some irregularities when grains touch each other.

Zhang and Liu [ZL] described an ellipse detector that may be used for real-time face detection. They first extract edges from the image using a robust edge detector. The center of an ellipse is detected by intersecting two lines that pass through the intersection of tangent lines and the midpoint of the corresponding chord. The parameter space of the Hough transform is decomposed to achieve computational efficiency.

Chapter 3 Computing shape orientation of point sets

This chapter will include several new methods for finding shape orientations. All of them are boundary based.

3.1 Moment Based Orientation

So far, the application of second degree moments to finding orientation has strictly been defined for areas of a shape. We have however tried applying the same orientation finding, moment based formulas for just the perimeter pixels of input shapes, and have found that similar results are obtained. Finding the orientation of shapes using just their perimeters is a novelty here as far as we are aware. The actual performance of the new orientation based methods versus the traditional area based one is left as future work. The formula used here to find the orientation of shape boundaries is the same as the standard area moment based one:

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

It was observed that the formula for the orientation line sometimes produces a line that is orthogonal to the desired one. We prove that the actual orientation is either the one from the standard moment based formula, or its orthogonal line, and use this fact in several linearity measures. 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 which are used in our experiments.

All definitions are applicable on sets of points with real coordinates. Moments, however, are also defined for infinite sets of points, such as open or closed curves, or for all points located inside a closed curve. Moments are typically used on closed curves, where all points within the closed curve are considered in their calculation.

3.2 Average α Orientation

This method takes k pairs of random points (p_1, p_2) from the shape and finds the angles α that these lines p_1p_2 make with the x-axis. The orientation of the shape is the median α value. In Figure 3.1, we see angles, α_1, α_2, α_3, which correspond to the green, brown and blue line segment angles with respect to the x-axis, respectively. The x-axis is represented by the light grey lines in this figure. The median value of the angles represents the orientation of the shape.

Figure 3.1 Average α illustrated on three line segments

3.3 Polygonal Shape Orientation for Manyfold Rotationally Symmetric Shapes

In this section we define a new method for polygonal shapes orientation. We do not restrict ourselves to polygonal shapes. In most situations, the equations (or other precise descriptions) of shape boundaries remain unknown. In order to preserve easier data manipulation, shape boundaries are usually piecewise approximated with straight line segments, conic arcs, or spline arcs. Approximating the boundary by a number of straight line segments (i.e. a use of polygonal approximation) is algorithmically simplest and computationally fastest. Another argument for using polygonal approximations is that there are a variety of good algorithms for polygonal approximations of discrete shapes (for an overview see [R4]).

Many shapes are polygonal by nature: machine made products, crystals, house outlines on satellite images, etc. Also, many shapes can be efficiently approximated by polygonal shapes that have relatively few edges. In the initial version given by Definition 3.3.1, the method has some undesirable properties. One of them is that there are situations where the method does not work. Many-fold rotationally symmetric shapes are shapes that could not be oriented if the initial version is applied. A modification of the method is proposed in order to overcome the apparent problems.

Zunic [Z] already presented examples which illustrate that the *ZO* method gives reasonable orientations. On the other hand, he already illustrated that there are some situations where the new method does not work. It has been already shown that the function $F(\alpha, P)$ is a constant function if P is a regular triangle or if P is a square. This implies that no direction could be pointed out as the shape orientation. There are many shapes, regular and irregular, whose orientation cannot be determined by *ZO*. A particular class of shapes whose orientation cannot be computed by *ZO* is the class of M-fold rotationally symmetric shapes ($M > 2$). The next lemma shows that for each polygonal shape having more than two axes of symmetry and more generally, each M-fold rotationally symmetric (with $M > 2$) shape P, the function $F(\alpha, P)$ is constant and consequently the shape orientation of P cannot be computed by the new method.

Lemma 3.3.1 If a polygonal shape with boundary P is M-fold rotationally symmetric and if $M > 2$ then $F(\alpha, P)$ is a constant function.

Proof - The proved equality in theorem 2.2.1 shows that there are only 4 extreme points (two maxima and two minima) of the function $F(\alpha, P)$ on the interval $[0, 2\pi)$. If P is M-fold rotationally symmetric with $M > 2$ then the function $F(\alpha, P)$ must have (at least) one minima and one maxima at each interval of the form

$$\left[(i-1)\cdot\frac{2\pi}{M}, i\cdot\frac{2\pi}{M}\right), \quad i = 1,\ldots,M$$

or it is a constant function. Because $M > 2$ is assumed there cannot be $2M$ strict extreme points, which implies that $F(\alpha, P)$ is constant. That establishes the proof.♦

The result of the previous lemma is not a surprise. A similar situation occurs is if the standard method is applied – for details see [TC, ZKF]. Due to its simplicity, the function $F(\alpha, P)$ is not a strong enough mathematical tool that can be used to define the orientation for many-fold rotationally symmetric shapes. It is worth mentioning that it is not expected

for the orientation of M-fold rotationally symmetric shapes to be uniquely defined. The most natural result is to have M directions that concurrently define the shape orientation. The mutual angles between these directions should be $i \cdot 2\pi/M$, where $i = 1, \ldots, M$. Thus, if we would like to have a computable orientation for a wider shape class (that includes rotationally symmetric shapes) the method should be modified. The replacement of the exponent 2 in definition 3.3.1 with a bigger number seems to be a natural modification. We give the following definition.

Definition 3.3.1 Let P be a shape with a polygonal boundary. Then, the shape orientation of P is defined by the angle α such that the total sum

$$F_{2N}(\alpha, P) = \sum_{e_i \text{ is an edge of } P} |pr_{\overline{\alpha}}(e_i)|^{2N} = \sum_{e_i \text{ is an edge of } P} |e_i|^{2N} \cos^{2N}(\alpha_i - \alpha)$$

of $2N$-powers of the lengths of the projections of the edges of P onto a line having the slope α is the maximal possible. The edges of shape P are represented by e_i, $i = 1, \ldots, n$. Recall that α_i (for $i = 1, \ldots, n$) denotes the angle between e_i and the x-axis.

The "complexity" of $F_{2N}(\alpha, P)$ increases with an increase of N. In order to have $F_{2N}(\alpha, P)$ = constant the polygon P has to satisfy stronger criteria than in the case of $N = 2$. But due to the diversity of the shapes involved, as big exponent $2N$ is chosen it is always possible to find a polygon P such that $F_{2N}(\alpha, P)$ is constant, and consequently, the method cannot be used for the computation of the orientation of P. Particularly, M-fold rotationally symmetric shapes with $M > 2N$ are such shapes, as stated by the following theorem.

Theorem 3.3.1 Let P be the boundary of an M-fold rotationally symmetric polygonal shape. Then $F_{2N}(\alpha, P)$ is constant if $2N < M$.

Proof. Let P be an M-fold rotationally symmetric polygon. First, we will derive that there are not more than $4N$ values of α for which $dF_{2N}(\alpha, P)/d\alpha$ vanishes. Indeed, starting from e_i is and edge of P

$$\frac{dF_{2N}(\alpha, P)}{d\alpha} = \frac{d\left(\sum_{e_i \text{ is an edge of } P} |e_i|^{2N} (\cos\alpha_i \cos\alpha + \sin\alpha_i \sin\alpha)^{2N} \right)}{d\alpha} =$$

$$\sum_{e_i \text{ is an edge of } P} 2N \cdot |e_i|^{2N} (\cos\alpha_i \cos\alpha + \sin\alpha_i \sin\alpha)^{2N-1} \cdot (-\cos\alpha_i \sin\alpha + \sin\alpha_i \cos\alpha)$$

we distinguish two situations (denoted by (i) and (ii)) depending on the values of $dF_{2N}(\alpha = 0, P)/d\alpha$ and $dF_{2N}(\alpha = \pi, P)/d\alpha$.

(i) If $\alpha = 0$ and $\alpha = \pi$ (i.e. $\sin\alpha = 0$) are not solutions of $dF_{2N}(\alpha, P)/d\alpha = 0$, then we have

$$\frac{dF_{2N}(\alpha, P)}{d\alpha} = 0 \Leftrightarrow$$

$$(\sin\alpha)^{2N} \cdot \sum_{e_i \text{ is an edge of } P} |e_i|^{2N} (\cos\alpha_i \cot\alpha + \sin\alpha_i)^{2N-1} \cdot (-\cos\alpha_i + \sin\alpha_i \cot\alpha) = 0$$

Since the quantity

$$\sum_{e_i \text{ is an edge of } P} |e_i|^{2N} (\cos\alpha_i \cot\alpha + \sin\alpha_i)^{2N-1} \cdot (-\cos\alpha_i + \sin\alpha_i \cot\alpha)$$

is a $2N$-degree polynomial on cot α it cannot have more than $2N$ real zeros

$$\cot\alpha_1 = z_1,\ \cot\alpha_2 = z_2,\ \ldots,\ \cot\alpha_k = z_k,\ (k \le 2N).$$

Since cot α=cot($\alpha+\pi$) we deduce that the equation $\frac{dF_{2N}(\alpha, P)}{d\alpha} = 0$ has no more than $4N$ solutions, assuming that $F_{2N}(\alpha, P)$ is not a constant function.

(ii) if $\alpha=0$ and $\alpha=\pi$, (i.e. sin $\alpha=0$), are solutions of $dF_{2N}(\alpha, P)/d\alpha = 0$, then

$$\sum_{e_i \text{ is an edge of } P} |e_i|^{2N} \cdot \cos^{2N-1}\alpha_i \cdot \sin\alpha_i = 0.$$

Then we get $(2N-1)$-degree polynomial in cot α since the coefficient (above expression on the left side) of $(\cot\alpha)^{2N}$ vanishes. Consequently, the polynomial cannot have more than $2N-1$ real zeros:

$$\cot\alpha_1 = z_1,\ \cot\alpha_2 = z_2,\ \ldots,\ \cot\alpha_k = z_k,\ (k \le 2N-1),$$

i.e. there are no more than $2(2N-1) = 4N-2$ values of α for which $dF_{2N}(\alpha, P)/d\alpha)$ vanishes. So, again, including $\alpha = 0$ and $\alpha = \pi$, the total number of zeros of $dF_{2N}(\alpha, P)/d\alpha=0$ is at most $4N$.

Thus, in both cases ((i) and (ii)) the number of zeros of $dF_{2N}(\alpha, P)/d\alpha$ is upper bounded by $4N$. On the other hand, if P is a fixed M-fold rotationally symmetric polygon, then $F_{2N}(\alpha, P)$ must have (because of the symmetry) at least M local minima and M local maxima (one minimum and one maximum at any interval of the form $[\alpha, \alpha+2\pi/M)$, or it must be a constant function. That means that $dF_{2N}(\alpha, P)/d\alpha$ must have (at least) $2M$ zeros $\alpha_1, \alpha_2, \ldots, \alpha_{2M}$. Since the presumption $2N < M$ does not allow $2M$ zeros of $F_{2N}(\alpha, P)/d\alpha$ if $F_{2N}(\alpha, P)$ is not a constant function, we just derived a contradiction. So, $F_{2N}(\alpha, P)$ must be constant whenever $2N < M$.♦

Several many fold rotationally symmetric shapes are presented in Fig 3.2, while their orientations computed based on $F_{2N}(\alpha, P)$ for $N = 1$, $N = 2$, $N = 3$, $N = 4$, and $N = 5$ are in Table 3.1. Note that for $N>2$, the computation of α is done by numerical analysis methods. The results are in accordance with Theorem 3.1.1. A regular triangle (Fig.4(a1)) cannot be oriented if $F_2(\alpha, P)$ and $F_4(\alpha, P)$ are used, while a use of $F_6(\alpha, P)$, $F_8(\alpha, P)$, and $F_{10}(\alpha, P)$ gives a 0^o orientation – of course the orientations 120^o and 240^o are congruent to the orientation of 0^o and $F_{2N}(0^o, P) = F_{2N}(120^o, P) = F_{2N}(240^o, P)$ holds in the presented case. It is worth mentioning that a modification of the standard method from [ZKF] gives, a perhaps more acceptable, orientation of 60 degrees. The shape in Fig 3.2(a2) is 4-fold rotationally symmetric, and consequently cannot be oriented if $F_2(\alpha, P)$ is used. A use of $F_4(\alpha, P)$, $F_6(\alpha, P)$, $F_9(\alpha, P)$, and $F_{10}(\alpha, P)$ leads to acceptable results. A similar discussion is valid for the last two shapes in Figure 3.2.

Figure 3.2 Orientation of manifold rotational symmetric shapes

Table 3.1 ORIENTATIONS AS COMPUTED BY APPLYING DEFINITION 3.1.1 TO THE SHAPES FROM FIG 3.2

	Fig 3.5(a1)	Fig 3.5(a2)	Fig 3.5 (a3)	Fig 3.5 (a4)
N=1	not computable	not computable	not computable	not computable
N=2	not computable	88°	not computable	not computable
N=3	0°	89°	not computable	172°
N=4	0°	90°	not computable	175°
N=5	0°	90°	0°	176°

The irregular shapes in Fig 3.3 are oriented by using $F_{2N}(\alpha, P)$ for different values of N. Shapes Fig 3.3(a1)-(a4) are almost reflectively symmetric and their computed orientations are in accordance with Lemma 3.1.1 (if $F_2(\alpha, P)$ is used). The shape in Fig 3.3(b2) seems to be easily orientable (i.e. it has a distinct orientation) and that is the reason that there are no big variations in the computed orientation when $2N$ changes. The rest of the results are acceptable.

Table 3.2 ORIENTATIONS COMPUTED FOR THE SHAPES FROM FIG 3.3

	Fig 3.6(a1)	Fig 3.6(a2)	Fig 3.6 (a3)	Fig 3.6 (a4)
N=1	91°	0°	89°	91°
N=2	91°	0°	90°	93°
N=3	91°	0°	90°	95°
N=4	92°	0°	90°	98°
N=5	92°	0°	90°	99°
	Fig 3.6(b1)	**Fig 3.6(b2)**	**Fig 3.6 (b3)**	**Fig 3.6 (b4)**
N=1	140°	103°	127°	114°
N=2	137°	103°	144°	123°
N=3	134°	102°	149°	133°
N=4	134°	102°	150°	135°
N=5	138°	102°	151°	135°
	Fig 3.6(c1)	**Fig 3.6(c2)**	**Fig 3.6 (c3)**	**Fig 3.6 (c4)**
N=1	44°	10°	96°	67°
N=2	46°	178°	94°	126°
N=3	48°	178°	96°	128°
N=4	48°	179°	98°	128°
N=5	48°	179°	100°	129°

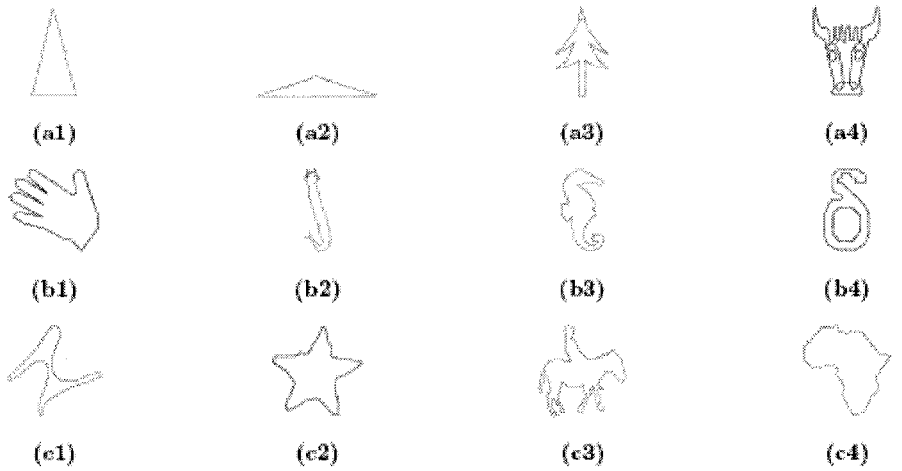


Figure 3.3 Shapes oriented using $F_{2N}(\alpha, P)$ for different N. The results are in Table 3.2

3.4 Orientation of Shapes with Arbitrary Boundaries

Another modification will be described in this section. It is motivated by the following question: *If a curve is replaced by a polygonal line whose vertices are sample points belonging to the curve, does the computed orientation (defined by Definition 2.2.1) converge when the maximum distance between consecutive points decrease to zero?* In other informal words, we consider whether the computed orientation of a curve (not necessarily a polygonal one) converges when the density of the curve sample points increases (see Fig 3.4 for an illustration).

We need some well-known mathematics to verify this. Taking into account trivial trigonometric identities

$$\sin(2\alpha)=\frac{2\tan\alpha}{1+\tan^2\alpha} \qquad \cos(2\alpha)=\frac{1-\tan^2\alpha}{1+\tan^2\alpha} \qquad (1)$$

and the fact that the first derivative $\dot{y}/\dot{x}=dx/dy$ of a curve given as $x = x(t)$ $y = y(t)$ equals the tangent of the angle between the curve tangent and the x-axis, we obtain the following identities (2) and (3) that hold for piecewise smooth enough curves, where $\dot{x} = dx/dt$, $\dot{y} = dy/dt$.

Lemma 3.4.1 Let C be a piecewise smooth enough curve given in parametric form $x = x(t)$, $y = y(t)$ where $t \in [a,b]$. Let

$$A_1 = (x(t = a), y(t = a)), A_2, \ldots, A_{k-1}, A_k = (x(t = b), y(t = b))$$

be points from the curve C, and let α_i denotes the angle between the $\overrightarrow{A_iA_{i+1}}$ and the x-axis. Then

$$\lim_{\substack{k\to\infty \\ \max\{|A_iA_{i+1}|,1\le i<k\}\to 0}} \sum_{i=1}^{k-1}|A_iA_{i+1}|\cdot\sin(2\alpha_i)=\oint_C\frac{2\dot{x}\dot{y}}{\dot{x}^2+\dot{y}^2}ds=\int_a^b\frac{2\dot{x}\dot{y}}{\sqrt{\dot{x}^2+\dot{y}^2}}dt \qquad (2)$$

and

$$\lim_{\substack{k\to\infty \\ \max\{|A_iA_{i+1}|,1\le i<k\}\to 0}} \sum_{i=1}^{k-1}|A_iA_{i+1}|\cdot\cos(2\alpha_i)=\oint_C \frac{\dot{x}^2-\dot{y}^2}{\dot{x}^2+\dot{y}^2}ds=\int_a^b \frac{\dot{x}^2-\dot{y}^2}{\sqrt{\dot{x}^2+\dot{y}^2}}dt. \qquad (3)$$

Thus if the number of the points on the curve increase such that the maximum distance between consecutive points tends to zero, both quantities $\sum_{i=1}^{k-1}|A_iA_{i+1}|\cdot\sin(2\alpha_i)$ and $\sum_{i=1}^{k-1}|A_iA_{i+1}|\cdot\cos(2\alpha_i)$ converge, as given by (2) and (3). Also, since

$$\left|\sum_{1\le i<k}|A_iA_{i+1}|^2\cdot\sin(2\alpha_i)\right|\le \max\{|A_iA_{i+1}|,1\le i<k\}\cdot\sum_{1\le i<k}|A_iA_{i+1}|\cdot\sin(2\alpha_i) \text{ i.e.}$$

$$\left|\sum_{1\le i<k}|A_iA_{i+1}|^2\cdot\cos(2\alpha_i)\right|\le \max\{|A_iA_{i+1}|,1\le i<k\}\cdot\sum_{1\le i<k}|A_iA_{i+1}|\cdot\cos(2\alpha_i)$$

and because of (2), (3), and $\max\{|A_iA_{i+1}|,\ 1\le i<k\}\to 0$ we have

$$\lim_{\substack{k\to\infty \\ \max\{|A_iA_{i+1}|,1\le i<k\}\to 0}} \sum_{i=1}^{k-1}|A_iA_{i+1}|^2\cdot\sin(2\alpha_i)=0, \qquad \lim_{\substack{k\to\infty \\ \max\{|A_iA_{i+1}|,1\le i<k\}\to 0}} \sum_{i=1}^{k-1}|A_iA_{i+1}|^2\cdot\cos(2\alpha_i)=0.$$

The quantities $\sum_{i=i}^{k-1}|A_iA_{i+1}|^2\cdot\sin(2\alpha_i)$ and $\sum_{i=i}^{k-1}|A_iA_{i+1}|^2\cdot\cos(2\alpha_i)$ actually appear in Theorem 2.2.1 and their ratio equals the tangent of the double angle that is the assigned orientation to the polygonal line L determined by the sample points. The convergence of such a ratio

$$\frac{\sum_{i=1}^{k-1}|A_iA_{i+1}|^2\cdot\sin(2\alpha_i)}{\sum_{i=1}^{k-1}|A_iA_{i+1}|^2\cdot\cos(2\alpha_i)}$$

is not guarantied in a general case. That is illustrated by Fig 3.4: Fifty points x_i are selected at random three times. The polygonal line having vertices $(0,0)=(x_1,x_1^2),(x_2,x_2^2),\ldots,(x_{49},x_{49}^2),(x_{50},x_{50}^2)=(1,1)$ is oriented by the new method. The following orientations are obtained:

Fig 3.4(a1) The computed orientation is 40 degrees. The abscissas of the selected points are:
0, 0.0154898, 0.0376846, 0.171443, 0.171633, 0.257857, 0.271856, 0.278504, 0.28133, 0.301993, 0.305018, 0.36028, 0.372977, 0.402503, 0.428349, .440952, 0.444447, 0.448548, 0.4561, 0.46496, 0.527588, 0.54522, 0.547799, 0.579116, 0.595214, 0.60459, 0.6282, 0.652748, 0.681778, 0.688673 0.704557, 0.70498, 0.72172, 0.72928, 0.78194, 0.80431, 0.818802, 0.818851, 0.828335, 0.842861, 0.864481, 0.877977, 0.885903, 0.897923, 0.905867, 0.914935, 0.926509, 0.932622, 0.968034, 1.

Fig 3.4 (a2) The computed orientation is 48 degrees. The abscissas of the selected points are:
0, 0.0195869, 0.0369824, 0.041192, 0.0977134, 0.14122, 0.16451, 0.194032, 0.227913, 0.255484, 0.260898, 0.274971, 0.32180, 0.34740, 0.35739, 0.36247, 0.41977, 0.431085, 0.437637, 0.461827, 0.464672, 0.52494, 0.54177, 0.580738, 0.58357, 0.58912, 0.59408, 0.645111, 0.653701, 0.664374, 0.67948, 0.71158, 0.71524, 0.71797, 0.75207, 0.752337, 0.768859, 0.772919, 0.792833, 0.793663,

0.803364, 0.845768, 0.864918, 0.905723, 0.910777, 0.912603, 0.939602, 0.961225, 0.968322, 1.

Fig 3.4 (a3) The computed orientation is 56 degrees. The abscissas of the selected points are: 0, 0.0197721, 0.0435937, 0.0660177, 0.115221, 0.131257, 0.149599, 0.152774, 0.157031, 0.16493, 0.180587, 0.20863, 0.21061, 0.24919, 0.25102, 0.25721, 0.260619, 0.262162, 0.285536, 0.293278, 0.325444, 0.37996, 0.39886, 0.39959, 0.43143, 0.48520, 0.494069, 0.496152, 0.504894, 0.507121, 0.5197, 0.5229, 0.52722, 0.532153, 0.539999, 0.54289, 0.572923, 0.578154, 0.600738, 0.615707, 0.616672, 0.624058, 0.664661, 0.665486, 0.695302, 0.703837, 0.828887, 0.836907, 0.883417, 1.

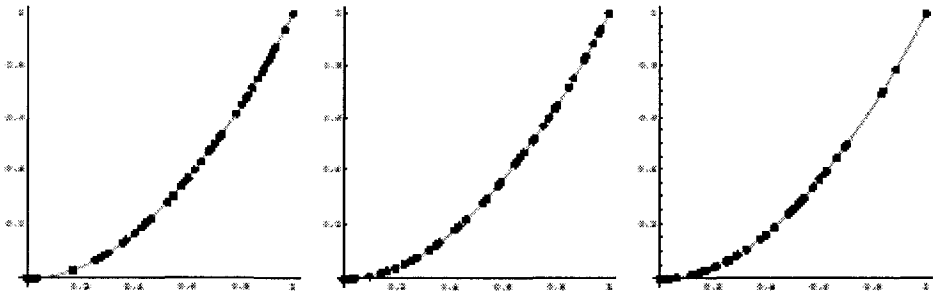


Figure 3.4 Fifty randomly selected points $\left(x_i x_i^2\right)$ are displayed

The polygonal line $(0,0)=\left(x_1,x_1^2\right),\left(x_2,x_2^2\right),\ldots,\left(x_{49},x_{49}^2\right),\left(x_{50},x_{50}^2\right)=(1,1)$ is oriented by the new method. The following orientations are obtained: (a1) 40 degrees; (a2) 48 degrees; (a3) 56 degrees.

Starting from the inequalities

$$\frac{\min\left\{\left|A_iA_{i+1}\right|,1\le i<k\right\}}{\max\left\{\left|A_iA_{i+1}\right|,1\le i<k\right\}}\cdot\frac{\sum\limits_{1\le i<k}\left|A_iA_{i+1}\right|\sin(2\alpha_i)}{\sum\limits_{1\le i<k}\left|A_iA_{i+1}\right|\cos(2\alpha_i)}\le\frac{\sum\limits_{1\le i<k}\left|A_iA_{i+1}\right|^2\sin(2\alpha_i)}{\sum\limits_{1\le i<k}\left|A_iA_{i+1}\right|^2\cos(2\alpha_i)} \tag{4}$$

and

$$\frac{\sum\limits_{1\le i<k}\left|A_iA_{i+1}\right|^2\sin(2\alpha_i)}{\sum\limits_{1\le i<k}\left|A_iA_{i+1}\right|^2\cos(2\alpha_i)}\le\frac{\max\left\{\left|A_iA_{i+1}\right|,1\le i<k\right\}}{\min\left\{\left|A_iA_{i+1}\right|,1\le i<k\right\}}\cdot\frac{\sum\limits_{1\le i<k}\left|A_iA_{i+1}\right|\sin(2\alpha_i)}{\sum\limits_{1\le i<k}\left|A_iA_{i+1}\right|\cos(2\alpha_i)} \tag{5}$$

we can see that the convergence of the computed orientations is guarantied if

$$\left|A_iA_{i+1}\right|=\left|A_jA_{j+1}\right| \text{ for all } i,j \text{ from } \{1, 2, \ldots, k\text{-}1\} \tag{6}$$

holds. In such a case $\frac{\min\left\{\left|A_iA_{i+1}\right|,1\le i\le k\right\}}{\max\left\{\left|A_iA_{i+1}\right|,1\le i\le k\right\}}=\frac{\max\left\{\left|A_iA_{i+1}\right|,1\le i\le k\right\}}{\min\left\{\left|A_iA_{i+1}\right|,1\le i\le k\right\}}=1$ and consequently (by (2)-(5)) the computed orientation converges to a fixed value α that satisfies

$$\tan(2\alpha)=\frac{\int_a^b \frac{2\dot{x}\dot{y}}{\sqrt{\dot{x}^2+\dot{y}^2}}dt}{\int_a^b \frac{\dot{x}^2-\dot{y}^2}{\sqrt{\dot{x}^2+\dot{y}^2}}dt}. \quad (7)$$

In this equation, $\dot{x}=dx/dt$ and $\dot{y}=dy/dt$. If (6) is not preserved we have the following estimate for computed orientations

$$\frac{1}{K}\cdot L \le \tan(2\alpha) \le K\cdot L \quad (8)$$

where $K=\frac{\max\{|A_iA_{i+1}|, 1\le i\le k\}}{\min\{|A_iA_{i+1}|, 1\le i\le k\}}$ and $L=\frac{\int_a^b \frac{2\dot{x}\dot{y}}{\sqrt{\dot{x}^2+\dot{y}^2}}dt}{\int_a^b \frac{\dot{x}^2-\dot{y}^2}{\sqrt{\dot{x}^2+\dot{y}^2}}dt}$.

The problem is that it is very difficult (almost impossible in real applications) to preserve (6) to be satisfied. On the other hand, the described convergence would be a very nice property if working with shapes that are not polygonal by their nature or when working with shapes that do not have a simple (but reasonably good) polygonal representation.

In order to overcome such problems we define another shape orientation which is actually a modification of the method *ZO* from definition 2.2.1.

Definition 3.4.1 Let P be a shape with a polygonal boundary. The shape orientation is defined by the angle α for which the total sum

$$G(\alpha,P)=\sum_{e \text{ is an edge of } P}\frac{|\mathrm{pr}_{\bar{\alpha}}(e)|^2}{|e|}=\sum_{e \text{ is an edge of } P}|e|\cdot\cos^2(\alpha_i-\alpha) \quad (19)$$

reaches the maximum.

Table 3.3 shows the angles of orientations αA of shapes if they are oriented in accordance with Definition 3.4.1. The angles of orientations αZ and αA obtained from Definition 2.2.1 and Definition 3.4.1 perform similarly on the shapes presented on Fig 3.2-3.3. A general conclusion that can be made is that the impact of the long edges is smaller in the case of a use of αA – which is natural since the edge lengths participate as a linear term in Definition 3.4.1 while they participate as a square term in Definition 2.2.1. It is very interesting to notice that in all the cases presented in Fig 3.4 the orientations computed by Definition 3.4.1 are consistent and all are equal to 46° – which illustrates the "convergence" property of this method. That can be understood as an advantage with respect to the computations based on Definition 2.2.1 which give divergent orientations 40°, 48°, and 55°.

Table 3.3 SHAPE ORIENTATIONS αA COMPUTED BY USING DEFINITION 3.3.1.

Fig 3.3(a1)	Fig 3.3(a2)	Fig 3.3(a3)	Fig 3.3(a4)	Fig 3.3(a5)	Fig 3.3(a6)
100°	106°	164°	119°	107°	130°
Fig 3.3(b1)	Fig 3.3(b2)	Fig 3.3(b3)	Fig 3.3(b4)	Fig 3.3(b5)	Fig 3.3(b6)
91°	93°	89°	89°	100°	40°
Fig 3.4(a1)	Fig 3.4(a2)	Fig 3.4(a3)	Fig 3.4(a4)	Fig 3.4(a5)	
0°	175°	171°	175°	135°	
Fig 3.5(a1)	Fig 3.5(a2)	Fig 3.5(a3)	Fig 3.5(a4)		
not comp.	not comp.	not comp.	not comp.		
Fig 3.6(a1)	Fig 3.6(a2)	Fig 3.6(a3)	Fig 3.6(a4)		
91°	0°	89°	90°		
Fig 3.6(b1)	Fig 3.6(b2)	Fig 3.6(b3)	Fig 3.6(b4)		
142°	102°	124°	112°		
Fig 3.6(c1)	Fig 3.6(c2)	Fig 3.6(c3)	Fig 3.6(c4)		
44°	22°	100°	40°		
Fig 3.7(a1)	Fig 3.7(a2)	Fig 3.7(a3)			
46°	46°	46°			

A few more examples are given in Fig 3.5. Shapes (a1)-(a4) are by their nature presented by open lines. The last shape (a5) illustrates the application on a shape whose boundary is not completely extracted. The orientations for all shapes in Figure 3.5 are calculated by Definition 3.4.2 and by the standard method.

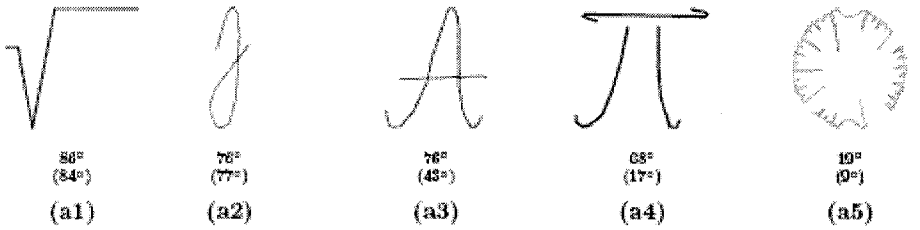


Figure 3.5 Shapes oriented by Def. 3.4.2 and by the standard method (in brackets)

Based on the previous discussion we prefer the orientation αA compared to the orientation αZ. The proven "convergence" property is the only reason for our choice.

Now, we give the following theorem that summarises the useful properties of αA. The proofs are omitted because of the analogy to the already proven statements in [Z].

Theorem 3.4.1 Let C be a piecewise smooth enough curve given in parametric form $x = x(t)$, $y = y(t)$ where $t \in [a,b]$. Let $A_1 = (x(t = a), y(t = a))$, $A_2, \ldots, A_{k-1}$, $A_k = (x(t = b), y(t = b))$ be points from the curve C while α_i denotes the angle between the edge $e_i = [A_iA_{i+1}]$ and the x-axis.

If $A_1, A_2, \ldots, A_k$ are consecutive vertices of a polygonal line P (not necessarily closed) then:

(a) The angle α for which the function $G(\alpha, P) = \sum_{e_i \text{ is an angle of } P} \frac{|\mathrm{pr}_{\bar{a}}(e_i)|^2}{|e_i|}$ reaches its maximum satisfies

$$\tan(2\alpha) = \frac{\sum_{e_i \text{ is an angle of } P} |e_i| \cdot \sin(2\alpha_i)}{\sum_{e_i \text{ is an angle of } P} |e_i| \cdot \cos(2\alpha_i)}$$

(b) If P is reflectively symmetric then the orientation αA computed by the method defined by Definition 3.4.1 is coincident with one of the symmetry axes of P or it is orthogonal to one of them;

(c) If P is M-fold rotationally symmetric, with $M > 2$, then $G(\alpha, P)$ is a constant function and it cannot be used to compute the orientation of P;

(d) Let the number k of sample points A_i from the curve C increase such that $max\{|A_iA_{i+1}|, 1 \le i < k\} \to 0$. If P_k is the polygon line determined by the vertices $A_1, A_2, \ldots, A_k$ and if α_k is the point where the function $G(\alpha, P_k)$ reaches the maximum then

$$\max\{|A_iA_{i+1}|, 1 \le i \le k\} \to 0 \quad \lim_{k\to\infty} \tan(2\overline{\alpha}_k) = \frac{\int_a^b \frac{2\dot{x}\dot{y}}{\sqrt{\dot{x}^2+\dot{y}^2}}\,dt}{\int_a^b \frac{\dot{x}^2-\dot{y}^2}{\sqrt{\dot{x}^2+\dot{y}^2}}\,dt}$$

Because of item (d) of Theorem 3.4.1 it is natural to extend the applicability of the orientation αA to the class of shapes with a smooth enough boundary (not necessarily polygonal) in the following way.

Definition 3.4.2 Let C be a shape with its boundary given in a parametric form: $x = x(t)$, $y = y(t)$ while $t \in [a,b]$. Then, the orientation αA of C is given by

$$\tan(2 \cdot \alpha A) = \frac{\int_a^b \frac{2\dot{x}\dot{y}}{\sqrt{\dot{x}^2+\dot{y}^2}}\,dt}{\int_a^b \frac{\dot{x}^2-\dot{y}^2}{\sqrt{\dot{x}^2+\dot{y}^2}}\,dt}$$

Remark 3.4.1 If C is a polygonal line (with edges e_i and the corresponding angles α_i ($1 \le i \le n$)), it can be also represented in the form $x = x(t)$, $y = y(t)$, $t \in [a,b]$. It is important to notice that the orientation of C can be computed by applying the above formulas such that

$$\tan(2 \cdot \alpha A) = \frac{\sum_{i=1}^{n} |e_i \cdot \sin(2\alpha_i)|}{\sum_{i=1}^{n} |e_i \cdot \cos(2\alpha_i)|} = \frac{\int_a^b \frac{2\dot{x}\dot{y}}{\sqrt{\dot{x}^2+\dot{y}^2}}\,dt}{\int_a^b \frac{\dot{x}^2-\dot{y}^2}{\sqrt{\dot{x}^2+\dot{y}^2}}\,dt}$$

holds. That is the reason why we use the same notation αA in both Definition 3.4.1 and 3.4.2.

We proceed with an example which illustrates the application of Definition 3.4.2 and item (d) of Theorem 3.4.1 to a parabolic arc. Let us consider a parabolic arc $C(v)$ given by $y = x^2$ and $x \in [0, v)$. In accordance with Definition 3.4.2, and by using the parameterization $x = t$, $y = t^2$, $t \in [0, v)$, the orientation αA of such an arc satisfies:

$$\tan(2\alpha A) = \frac{\int_0^v \frac{2\dot{x}\dot{y}}{\sqrt{\dot{x}^2+\dot{y}^2}}dt}{\int_0^v \frac{\dot{x}^2-\dot{y}^2}{\sqrt{\dot{x}^2+\dot{y}^2}}dt} = \frac{\int_0^v \frac{4t}{\sqrt{1+4t^2}}dt}{\int_0^v \frac{1-4t^2}{\sqrt{1+4t^2}}dt} = \frac{\sqrt{1+4v^2}-1}{\frac{3}{4}\ln\left(2v+\sqrt{1+4v^2}\right)-\frac{1}{2}v\sqrt{1+4v^2}}$$

The graph of the function $G(v)$

$$\frac{\sqrt{1+4v^2}-1}{\frac{3}{4}\ln\left(2v+\sqrt{1+4v^2}\right)-\frac{1}{2}v\sqrt{1+4v^2}}$$

is in Figure 3.6 (a2). The limit values

$$\lim_{v\to 0}\frac{\sqrt{1+4v^2}-1}{\frac{3}{4}\ln\left(2v+\sqrt{1+4v^2}\right)-\frac{1}{2}v\sqrt{1+4v^2}} = \lim_{v\to\infty}\frac{\sqrt{1+4v^2}-1}{\frac{3}{4}\ln\left(2v+\sqrt{1+4v^2}\right)-\frac{1}{2}v\sqrt{1+4v^2}} = 0$$

are in accordance with our expectations: For very small v (close to zero) the arc C has orientation very close to 0° but also that for a very large v the orientation is close to 90°.

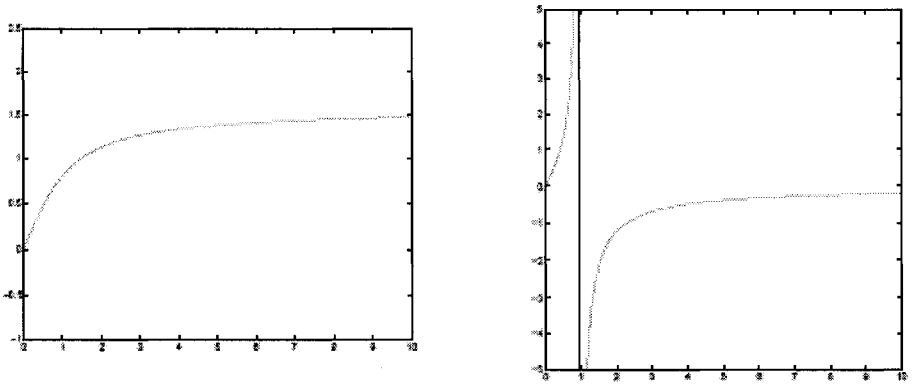
Furthermore, the function

$$\frac{\sqrt{1+4v^2}-1}{\frac{3}{4}\ln\left(2v+\sqrt{1+4v^2}\right)-\frac{1}{2}s\sqrt{1+4v^2}}$$

has a discontinuity at $v_0 \approx 0.9799$ that corresponds to the value $v = v_0$ for which the arc $C(v_0)$ has orientation $\pi/4$ (notice that then $lim_{\varepsilon\to\pm 0}\ \tan(2\cdot(C(v_0-\varepsilon))) = \pm\infty$). Since the standard definition of the function $\arctan(x)$ maps $(-\infty, \infty) \to (-\pi/2, \pi/2)$ the orientation $\alpha A(C(v))$ should be computed as

$$\alpha A(C(v)) = \begin{cases} \frac{1}{2}\arctan\left(\frac{\sqrt{1+4v^2}-1}{\frac{3}{4}\ln\left(2v+\sqrt{1+4v^2}\right)-\frac{1}{2}s\sqrt{1+4v^2}}\right) & \text{if } v\in(0,v_o) \\ \frac{\pi}{2}+\frac{1}{2}\arctan\left(\frac{\sqrt{1+4v^2}-1}{\frac{3}{4}\ln\left(2v+\sqrt{1+4v^2}\right)-\frac{1}{2}s\sqrt{1+4v^2}}\right) & \text{if } v\in(0,\infty). \end{cases}$$

The graph of the function $\alpha A(C(v))$ is in Fig 3.6(a1).



(a1) (a2)

Figure 3.6 (a1) The graph of the orientation αA. (a2) The graph of function G(v)

In the future work section (chapter 8), we will discuss the limitation of the proposed methods. The convergence property is limited to differentiable curves in R^2, and does not generalize to digital images. Furthermore, we will show that the *ZO* method [Z] and orientation methods for manyfold rotationally symmetric polygons and arbitrary curves presented here in chapter 3 appear to find local orientations of parts of shapes which may differ from the global, and perhaps natural, shape orientation.

3.5 Discussion on boundary projection based orientation methods

The projection based methods of calculating orientation have the inherent flaw of finding local orientation measures for parts of a shape instead of the orientation of the entire shape. For an illustration of this point, we turn to Figure 3.7.

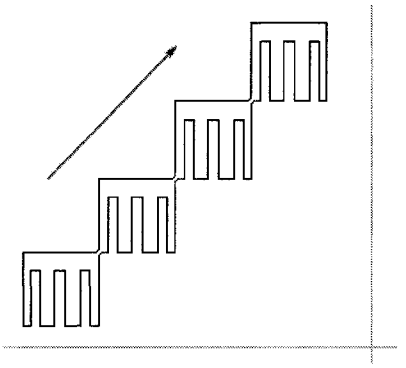


Figure 3.7 Shape that is not correctly oriented by the projections methods

This figure shows a shape that would have an orientation parallel to the y axis. It is evident the true orientation of this shape is consistent with the blue arrow in the same figure. Similar counter examples highlight problems with the projection based elongation measures (section 4.1) as well. We have proposed two alternate methods for computing orientation to correct these problems in sections 3.1 and 3.2.

Chapter 4 Measuring shape elongation of polygons and arbitrary shapes

4.1 Elongation Measure for Polygonal Shapes

Following the idea of the standard method for measuring shape elongation we define the new elongation measure as the ratio of the maximum and minimum value of the function

$$F(\alpha, P) = \sum_{e \text{ is an edge of } P} |pr_{\bar{a}}(e)|^2$$

that has been used for computing the shape orientation (definition 2.2.1 [Z]).

Theorem 4.1.1 - Let P be a shape with a polygonal boundary. Then the new elongation measure of P can be expressed as

$$E(P) = \frac{\sum_{1 \le i \le n} |e_i|^2 + \sqrt{\left(\sum_{1 \le i \le n} |e_i|^2 \cos(2\alpha_i)\right)^2 + \left(\sum_{1 \le i \le n} |e_i|^2 \sin(2\alpha_i)\right)^2}}{\sum_{1 \le i \le n} |e_i|^2 - \sqrt{\left(\sum_{1 \le i \le n} |e_i|^2 \cos(2\alpha_i)\right)^2 + \left(\sum_{1 \le i \le n} |e_i|^2 \sin(2\alpha_i)\right)^2}}$$

where e_i $(1 \le i \le n)$ are edges of the boundary of P and α_i $(1 \le i \le n)$ are angles between the edges e_i and the x-axis.

Proof – The length of the projection $pr_{\bar{a}}(e)$ of the edge e_i onto a line that has slope α is

$$|pr_{\bar{a}}(e)| = |e_i \cos(\alpha_i - \alpha)|.$$

By using a simple trigonometric identity $\cos^2(\alpha) = 1 + \cos 2\alpha/2$ we can transform the optimising function $F(\alpha, P)$ into:

$$F(\alpha, P) = \frac{1}{2} \cdot \sum_{1 \le i \le n} |e_i|^2 + \frac{1}{2} \cdot \sum_{1 \le i \le n} |e_i|^2 (\cos(2\alpha_i)\cos(2\alpha) + \sin(2\alpha_i)\sin(2\alpha)).$$

As proved in [Z], the angle values for which $F(\alpha, P)$ reaches its minimum and maximum satisfy

$$\frac{\sin(2\alpha)}{\cos(2\alpha)} = \frac{\sum_{i=1}^{n} |e_i|^2 \sin(2\alpha_i)}{\sum_{i=1}^{n} |e_i|^2 \cos(2\alpha_i)}.$$

Now, using the trigonometric identities:

$$\sin(2\varphi) = \frac{\pm \tan(2\varphi)}{\sqrt{1 + \tan^2(2\varphi)}} \text{ and } \cos(2\varphi) = \frac{\pm 1}{\sqrt{1 + \tan^2(2\varphi)}},$$

we derive that $\cos(2\varphi)$ and $\sin(2\varphi)$ at the extreme points of $F(\alpha, P)$ can be expressed (together) as

$$\cos(2\varphi)=\frac{\pm\sum_{1\le i\le n}|e_i|^2\cos(2\alpha_i)}{\sqrt{\left(\sum_{1\le i\le n}|e_i|^2\cos(2\alpha_i)\right)^2+\left(\sum_{1\le i\le n}|e_i|^2\sin(2\alpha_i)\right)^2}}$$

$$\sin(2\varphi)=\frac{\pm\sum_{1\le i\le n}|e_i|^2\sin(2\alpha_i)}{\sqrt{\left(\sum_{1\le i\le n}|e_i|^2\cos(2\alpha_i)\right)^2+\left(\sum_{1\le i\le n}|e_i|^2\sin(2\alpha_i)\right)^2}}.$$

Entering the last two equalities into the formula for $F(\alpha,P)$, we derive that the minimum and maximum of $F(\alpha, P)$ can be expressed as

$$\frac{1}{2}\sum_{1\le i\le n}|e_i|^2+\sum_{1\le i\le n}|e_i|^2\cdot\frac{\pm\cos(2\alpha_i)\cdot\sum_{1\le i\le n}|e_i|^2\cos(2\alpha_i)}{\sqrt{\left(\sum_{1\le i\le n}|e_i|^2\cos(2\alpha_i)\right)^2+\left(\sum_{1\le i\le n}|e_i|^2\sin(2\alpha_i)\right)^2}}$$

$$+\sum_{1\le i\le n}|e_i|^2\cdot\frac{\pm\sin(2\alpha_i)\cdot\sum_{1\le i\le n}|e_i|^2\sin(2\alpha_i)}{\sqrt{\left(\sum_{1\le i\le n}|e_i|^2\cos(2\alpha_i)\right)^2+\left(\sum_{1\le i\le n}|e_i|^2\sin(2\alpha_i)\right)^2}}$$

or equivalently as

$$\frac{1}{2}\sum_{1\le i\le n}|e_i|^2\pm\frac{\left(\sum_{1\le i\le n}|e_i|^2\cos(2\alpha_i)\right)^2+\left(\sum_{1\le i\le n}|e_i|^2\sin(2\alpha_i)\right)^2}{\sqrt{\left(\sum_{1\le i\le n}|e_i|^2\cos(2\alpha_i)\right)^2+\left(\sum_{1\le i\le n}|e_i|^2\sin(2\alpha_i)\right)^2}}.$$

Thus, we derived that the maximum and minimum of $F(\alpha, P)$ are as follows: $\max\{F(\alpha,P)\,|\,\alpha\in[0,2\pi]\}=$

$$\frac{1}{2}\cdot\sum_{1\le i\le n}|e_i|^2+\frac{1}{2}\cdot\sqrt{\left(\sum_{1\le i\le n}|e_i|^2\cos(2\alpha_i)\right)^2+\left(\sum_{1\le i\le n}|e_i|^2\sin(2\alpha_i)\right)^2}$$

and $\min\{F(\alpha,P)\,|\,\alpha\in[0,2\pi]\}=$

$$\frac{1}{2}\cdot\sum_{1\le i\le n}|e_i|^2-\frac{1}{2}\cdot\sqrt{\left(\sum_{1\le i\le n}|e_i|^2\cos(2\alpha_i)\right)^2+\left(\sum_{1\le i\le n}|e_i|^2\sin(2\alpha_i)\right)^2}.$$

This establishes the proof.♦

Lemma 4.1.1 considers two properties that encompass the new elongation measure. The proof is omitted because it follows directly from the definitions.

Lemma 4.1.1 - The new elongation measure satisfies the following properties:

– $E(P) \in [1, \infty)$ for each polygonal shape P;

– $E(P)$ is invariant with respect to similarity transformations.

It is worth mentioning that the new elongation measure is valid for both open and closed polygons, as it considers the boundary of the polygonal shape. It can be applied to open polygonal lines, but also to the set of several polygonal lines. This enables the method to be applicable to shapes whose boundaries are not completely extracted. The reasons for an incomplete extracted boundary could be: the shape is partially overlaid, there are large similarities between background pixels and pixels belonging to the shape, etc.

4.2 Experiments

In the previous section we proposed a new shape elongation measure. It is naturally motivated and simple to compute. There is a closed formula that expresses the elongation of a given polygonal shape as a function of the boundary edges and angles that those edges made with the x-axis. It performs well in some canonical cases. For example, let us consider a rectangle $T(a)$ having edge lengths a and 1. In accordance with Theorem 4.1.1, its measured elongation is

$$E(T(a)) = \frac{1 + a + \sqrt{a^4 + 1}}{1 + a - \sqrt{a^4 + 1}},$$

which is acceptable. In the limit cases where $a \to \infty$ and $a \to 0$ the rectangle degenerates into a line segment while the measured elongations tend to infinity. This behaviour is expected, and in fact preferred. In the case of $a = 1$ the measured elongation is equal to 1. In this case the rectangle degenerates into a square which is a 4-fold rotationally symmetric shape. Problems arising when working with manyfold rotationally symmetric shapes are discussed in [KK, KR]. Next we give several shapes with their measured elongations. The new measure E is boundary based and it is more sensitive to noise or to boundary defects (e.g. intrusions on the boundary) than the standard measure Es. That is illustrated by the first two examples from Fig 4.1. There is an essential difference between the measured elongations if the new measure E is used. On the other hand, there is only a small difference if those shapes are measured by the standard elongation measure Es. Such "sensitivity" is not necessarily a disadvantage; particularly when working in high precision (inspection) tasks.

The last two shapes in Fig 4.1 illustrate how shape deformations could affect the measured elongation. In those examples the rankings given by E and Es are different.

An advantage of the method is that it can be applied to shapes whose boundary consists of several polygonal lines (see the fourth shape in Fig 4.1) or to shapes with missing parts on their boundaries (see the last example on Fig 4.2). The fourth shape in Fig 4.1 presents a square with a triangular hole. It has a measured elongation ES very close to one. It is not surprising, because results from [TC, ZKF] imply that all N-fold rotationally symmetric shapes (if $N > 2$) have the same, minimal possible, measured elongation which is equal to 1. Since the percentage of pixels that correspond to the triangular hole is relatively small, it

does not lead to an essential change in the measured elongation *ES*. If the new measure *E* is applied then the impact of the hole is more significant. That can be understood as a desirable property.

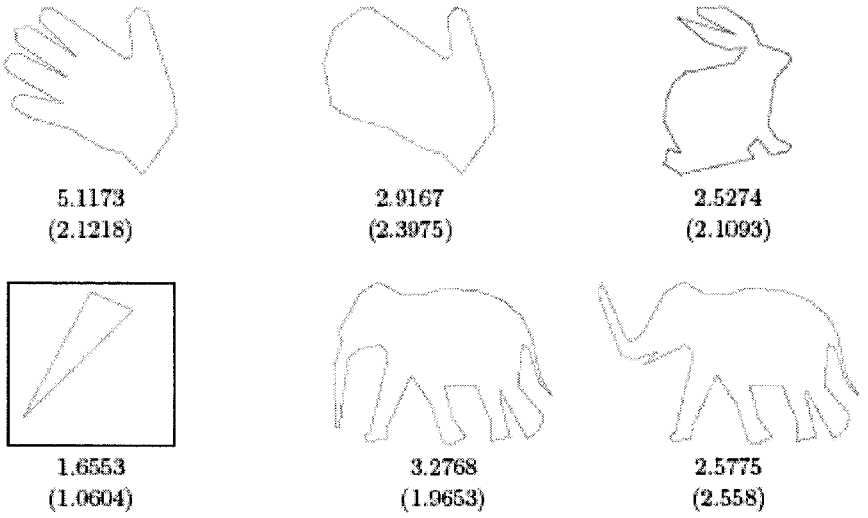


Figure 4.1 Elongations by the new method and by the standard method (in brackets)

Several shapes that are presented usually by a curved line (or several of them) are given in Fig 4.2.

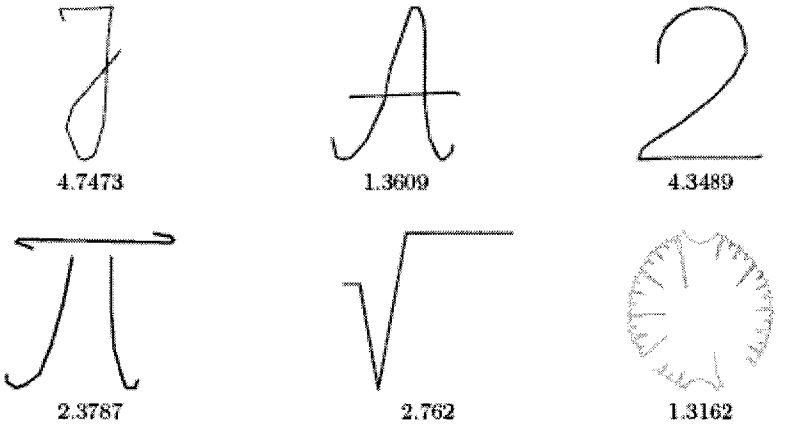


Figure 4.2 Computed elongations of polygonal lines by the new method.

The traditional shape elongation measure is area based. It is therefore defined only for closed shapes. In this article, we proposed a shape boundary based measure, with a closed formula. Using our new method, elongation can be measured for any open shape, including shapes composed of several components. The measure is invariant with respect to rotation, translation and scaling.

4.3 Elongation Measure for Arbitrary Shapes

In this section we extend the new elongation measure to shapes with arbitrary boundaries. Assume that we have a piecewise smooth enough curve p given in a parametric form $x = x(t)$, $y = y(t)$, ($t \in [a, b]$). The elongation $E(p)$ of the curve p is defined as

$$E(p)=\frac{Length(p)+\sqrt{\left(\int_a^b \frac{2\dot{x}\dot{y}}{\sqrt{\dot{x}^2+\dot{y}^2}}dt\right)^2+\left(\int_a^b \frac{\dot{x}^2-\dot{y}^2}{\sqrt{\dot{x}^2+\dot{y}^2}}dt\right)^2}}{Length(p)-\sqrt{\left(\int_a^b \frac{2\dot{x}\dot{y}}{\sqrt{\dot{x}^2+\dot{y}^2}}dt\right)^2+\left(\int_a^b \frac{\dot{x}^2-\dot{y}^2}{\sqrt{\dot{x}^2+\dot{y}^2}}dt\right)^2}}.$$

We have shown that the measure $E(P)$ satisfies the "convergence property" in chapter 4. Precisely, let us assume we have a curve and a set of sample points from it. Also, let us assume that we have the computed elongation $E(P)$ of the polygonal curve P whose vertices are the selected sample points. Then, roughly speaking, by the convergence property of an elongation measure we mean that the computed elongations (of polygonal curves determined by sample points) should converge when the density of sample points increases and the largest distance between any two consecutive sample points approaches zero. Naturally, if the convergence property holds, the limit value for the measured elongations of polygonal lines determined by sample points is used as the elongation measure $E(P)$ of the sampled curve.

As an illustration of the behaviour of shape elongation given by the definition for $E(p)$ we give the following synthetic example. Let us consider polygon $P(u)$ on a parabola segment defined by $y=x^2$ on the interval $[0, u)$, where u varies from 0 to infinity. The graph of $E(P(u))$ is presented in Fig.4.3(a). As it can be seen, for a very small u close to 0, the measured elongation is very high and tends to infinity as u tends to 0. After that, $E(P(u))$ decreases and reaches the minimum somewhere close to 1. $E(P(u))$ then increases again, and tends to infinity if u tends to infinity, too. Such behaviour is expected. Indeed, for a very small u, the parabola segment $P(u)$ is contained in a very elongated rectangle whose sides are u and u^2. If u becomes very big, then the rectangle that includes $P(u)$ is again very elongated because the ratio of its sides u^2 and u is again very big (see Fig 4.3(b)).

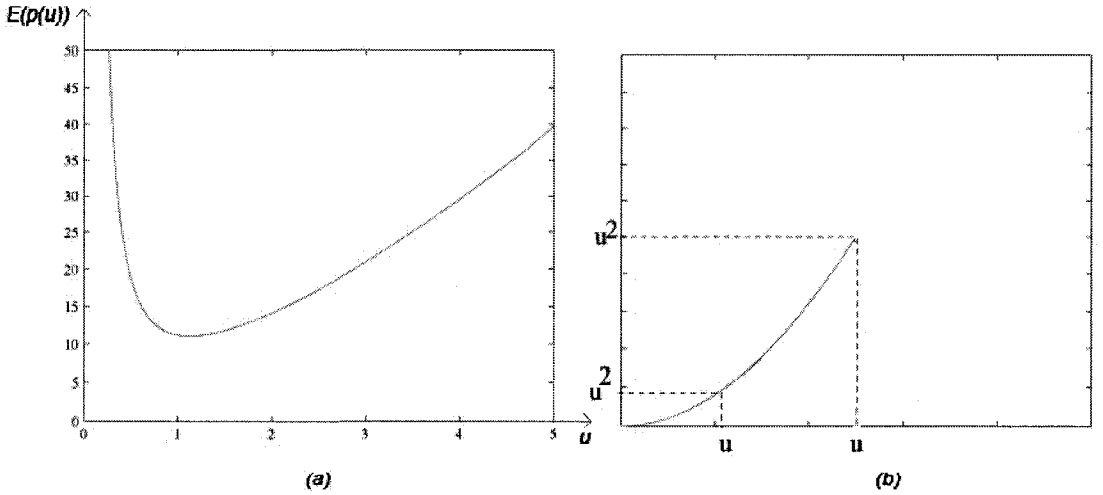


Figure 4.3 Graph of $E(P(u))$ an $E(P)$

4.4 Performance evaluation of boundary based elongation measures

We will now investigate the sensitivity of the new formula to the noise on the boundary. Generally speaking, all methods that use only boundary information must be sensitive to

boundary changes (e.g. deformations, intrusions, noise, etc). Inevitably, once we accept to work with boundary information and exploit the benefits that come from sensitive methods, we have also to accept problems which could come from boundary sensitivity. Typical problems are caused by noise on images that should be processed. Some of the problems when dealing with a noisy shape can be avoided by a suitable choice of polygonal approximation or by applying some standard procedures (e.g. smoothing). Some noise effects to the computed shape elongation are illustrated in Figure 4.4. It can be seen that minor noise could be acceptable for efficient elongation estimation as seen in Figure 4.4(*b*), while high noise could lead to a perceived inaccuracy such as in Figure 4.4(*d*).

1.4189 1.3729 1.2198 1.1799

(a) (b) (c) (d)

Figure 4.4 Computed elongations *E(P)* are given to illustrate the noise effects.

Another disadvantage of the method presented here could be derived from the fact that the new elongation measure depends on the edge lengths and the edge orientations but not on the order of edges. Let P be a polygonal boundary. Then the measured elongation $E(P)$ has the same value for all permutations of the edges of P.

Notice that if P is a closed polygon then all permutations of the edges of P do not necessarily give a closed polygon again (the result can be an open polygonal line or a self intersecting polygonal line), but the elongation of such resulting polygonal lines is still the same, which may contradict our perception. However, as far as we know, there is no created descriptor which reaches our perception in all situations. A generalization of the above statement follows.

Let $P = (e_1,\ldots, e_n)$ be a polygonal boundary, e_i an arbitrary, fixed edge of P and $f_1, \ldots, f_k$ a set of edges with the same slope as the slope of e_i and the total sum of edge lengths equals $|e_i|$ (i.e. $\sum_{1\le j\le k}|f_j| = |e_i|$). Then the elongation $E(P)$ equals the elongation of all the polygonal lines consisting of edges $\{e_l \mid 1\le l\le n, l\neq i\}\cup\{f_j \mid 1\le j\le k\}$.

The situation described above is illustrated by Figure 4.5. A digitalization [KR] of the south-east arc of a circle is presented in Figure 4.5(*b*). It consists of 19 vertical and 19 horizontal unit edges. If those edges are listed in the order: 19 horizontal edges first, after that 19 vertical edges, we get the polygonal line presented in Figure 4.5(*c*). Both polygonal lines have the same measured elongation equal to 1, which contradicts our perception.

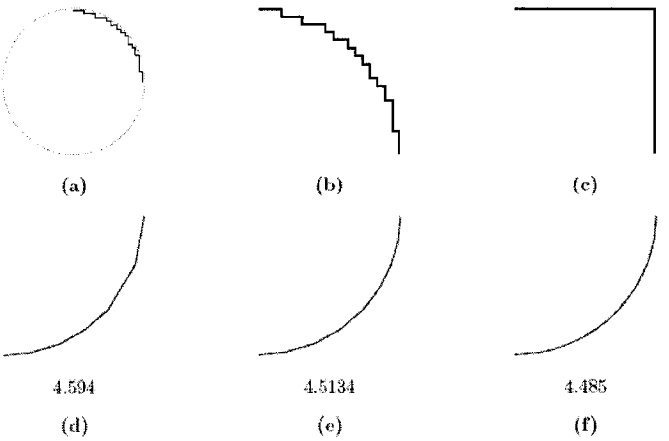


Figure 4.5 elongations E of shapes that consist of 19 vertical and 19 horizontal unit edges

Such an unpreferred elongation of shape in Figure 4.5(b) can be corrected if a suitable polygonal approximation of the presented digital arc is applied. In Figure 4.5 (d)-(f) three polygonal approximations are shown. The following computed elongations of circular arcs are obtained: 4.4595, 4.5134, and 4.485, respectively. It can be seen that the increase of slope accuracy of the approximation leads to the measured elongations which are closer to the theoretical value $(\pi+2)/(\pi-2) \approx 4.5039$ for the measured elongation of the south-east arc of a circle. This (exact) theoretical value $(\pi+2)/(\pi-2)$ is obtained by applying the formula from the definition of $E(P)$. Notice that any reasonably good polygonal approximation algorithm will keep the shape in Figure 4.5(c) almost unchanged and, consequently, the corresponding measured elongations would always be close to 1.

The projection based methods of calculating elongation have the inherent flaw of finding local elongation measures for parts of a shape instead of the elongation of the entire shape. For an illustration of this point, we turn to Figure 4.6.

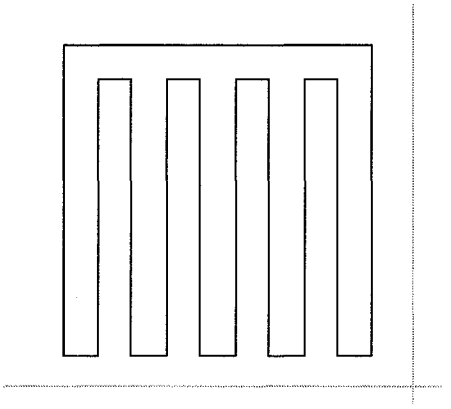


Figure 4.6 Shape whose elongation is not correctly calculated by the projections methods

This figure shows a shape that would have an extremely high elongation value assigned to it via the projections methods. This would be the case since it has many edges that are parallel to the y-axis. Therefore the computed ratio of y versus x axis projections would unfairly classify it as a very elongated shape. It is evident that the true orientation of this shape is more consistent to that of a square.

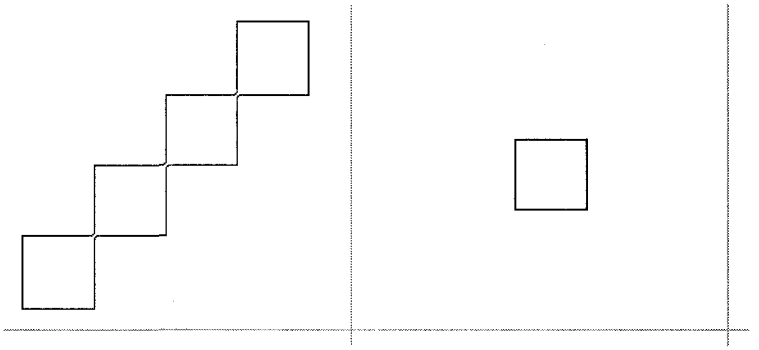


Figure 4.7 Shapes with identical elongations by the projections methods

The projection based elongation methods would give the same elongation measure to both of the shapes seen in Figure 4.7. This happens since the elongation of multiple copies of the same object, considering they all have the same orientation, equals the elongation of the original object. This is an inherently weakness of these methods.

Chapter 5 Measuring linearity of unordered point sets

The algorithms that we proposed and analyzed for measuring linearity of unordered point sets are described here. They are called: average orientations, rotation correlation or eccentricity, triangle heights, triangle perimeters, triplet 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 sufficiently accurate linearity measure, or for rejecting linearity of a set with sufficient confidence.

5.1 Average Orientations

Here, we first find the center of mass of the point set, and its angle of orientation using moments as described in section 3.1. The average orientation measure first 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 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. Please see Figure 5.1 below.

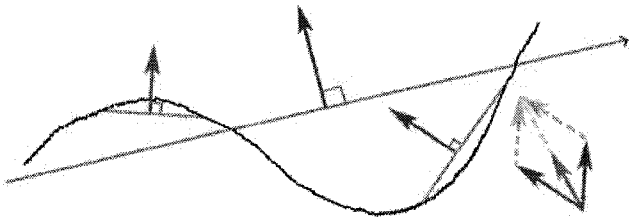


Figure 5.1 Normals all oriented in the same direction

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. Since the moment function sometimes produces an orientation line with an angle α which is 90° offset from what is visually the actual orientation line, we repeat the entire procedure for an angle $\alpha+\pi/2$, and select the higher of the two measures.

Algorithm: *average Orientations:*
Input: array of points: $Points=(X_i, Y_i)$, $1 \le i \le count$, 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 $M=\tan(\alpha)$ 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 $m=(y_2-y_1)/(x_2-x_1)$;
$a1=-m$, $b1=1$;
$dp=mM+1$;
Normalization factor $norm=\sqrt{(m^2+1)}$;
If ($dp<0$) **then** { $a[i]=(-a1/norm)$; $b[i]=(-b1/norm)$;}
else { $a[i]=(a1/norm)$; $b[i]=(b1/norm)$;}
}

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

$linearity = \sqrt{A^2 + B^2}$;
Repeat entire **for** loop for $a1=\alpha+\pi/2$, which results in a new linearity value, *linearity1*;
If (*linearity*< *linearity1*) **then** *linearity=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 Theorem 5.1.1. This is normalized to [0, 1]:

linearity= (linearity-2/π)/(1-2/π);
Output *linearity*;

In Lemma 5.1.1, 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 5.1.1 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, $1 \le i \le k$, 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

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

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

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

However,

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

$$= 4\sin^2(\theta/2)\Big|_0^{\pi/2} = 2.$$

Therefore the measure is $1/(\pi k)\sum_{i=1}^{k} 2 = 2/\pi$.♦

Theorem 5.1.1 The average orientation measure of arbitrary object is $\geq 2/\pi$, 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, $1 \leq i \leq k$, 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 $S = \int_0^{\pi} (\sum_{i=1}^{k} |\cos(\alpha_i-\theta)|/k\, d\theta < \pi 2/\pi = 2$. Thus $S = \sum_{i=1}^{k} (1/k) \int_0^{\pi} |\cos(\alpha_i-\theta)|\, d\theta < 2$. Therefore, there exist i such that $\int_0^{\pi} |\cos(\alpha_i-\theta)|\, d\theta < 2$. But $\int_0^{\pi} |\cos(\alpha_i-\theta)|\, d\theta = 2$ (see Lemma 5.1.1), which is a contradiction.♦

5.2 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. The procedure is repeated for the orientation line which is at an angle $\alpha 1 = \alpha + \pi/2$, where α is the original angle of orientation as determined by the moment function. The final output of the program is the greater of the two correlation values.

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

Find center of mass (X_c, Y_c) using moments;
Find angle of orientation α by the above formula;
$\alpha 1 = \alpha + \pi/2$;
$arot = \pi/4 - \alpha$; *//arot is angle of rotation of set of points to make them 45° to x axis.*

Rotate all points in the *Points* array by (*arot*) 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'), $1 \le i \le count$;

If (*Correlation* <0) **then** *Correlation=-Correlation*;

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

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

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

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

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

We will now prove the following theorem:

Theorem 5.2.1: 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 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.

5.3 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 k hc values as a linearity measure of the set of points. Figure 5.2 illustrates this point.

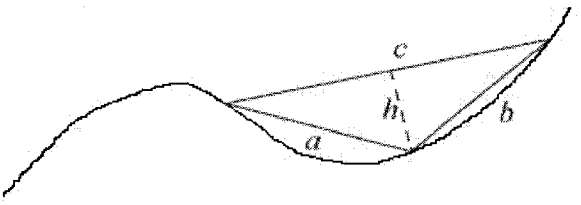


Figure 5.2 Triangle formed by 3 random points, and its height h

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:

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

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), $1 \le i \le count$, 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; *//*$a \le b \le c$
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 $h = (Au + Bv + e)/\sqrt{A^2 + B^2}$.
If (h<0) **then** $h=-h$;
$h=h/c$;

$h = 1 - \left(2h/\sqrt{3}\right)$

sumh=sumh+h;

}

linearity=sumh/k;

$linearity = (linearity - 0.66)/0.34$;

if *linearity*<0 **then** *linearity* = 0;

5.4 Triangle Perimeters

This method is similar to the previous one in the sense that we take k triplets of random points from the set of points 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 $a \le b \le c$. 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 close 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: $Points=(X_i, Y_i), 1 \le i \le count$, k;
Output: Linearity value;

Float *sump=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; $//a \le b \le c$
$p = (2c - a - b)/c$.
$sump = sump + p$;
}
linearity=sump/k;
$linearity = (linearity - 0.76)/0.24$
if *linearity*<0 **then** *linearity* = 0;

5.5 Triplet Smoothness

The triplet smoothness (*ts*) is inspired by the *contour smoothness* formula in [C] which is defined as $4\pi S/P^2$. In this formula, S is the area of the shape, and P is its perimeter. This formula is normally applied to measuring circularity. It bases its measurements on the area of a shape divided by the square of its perimeter. This measure was used as a compactness measure.

As opposed to the contour smoothness, 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: *triplet smoothness:*
Input: array of points: *Points*=(X_i, Y_i), $1 \le i \le count$, *k*;
Output: *Linearity* value;
sums=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*; // $a \le b \le c$
$p = a + b + c$;
Find area of the triangle formed by these 3 points: *area*;
*s=area/(p*p)*;
sums=sums+s;
}
$sums = (sums * 36)/\sqrt{3}$; //normalization factor
$linearity = (1 - sums - 0.45)/0.55$;
if *linearity*<0 **then** *linearity* = 0;

5.6 Ellipse Axis Ratio

Here, we use the idea of measuring rectangularity as proposed in [R2] (described in section 2.7), 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 *a* and minor *b* axis of the best fit ellipse as determined by the formulas in [R2] (see section 2.7). The linearity value is given as 1-*b/a*.

Algorithm: *Ellipse axis ratio:*
Input: array of points: *Points*=(X_i, Y_i), $1 \le i \le count$, *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;
Find value for minor axis b;
linearity = 1-b/a.

5.7 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 400x400 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 Figure 5.3. All 30 are examined by each algorithm. Their linearity values are presented in table 5.1. 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.

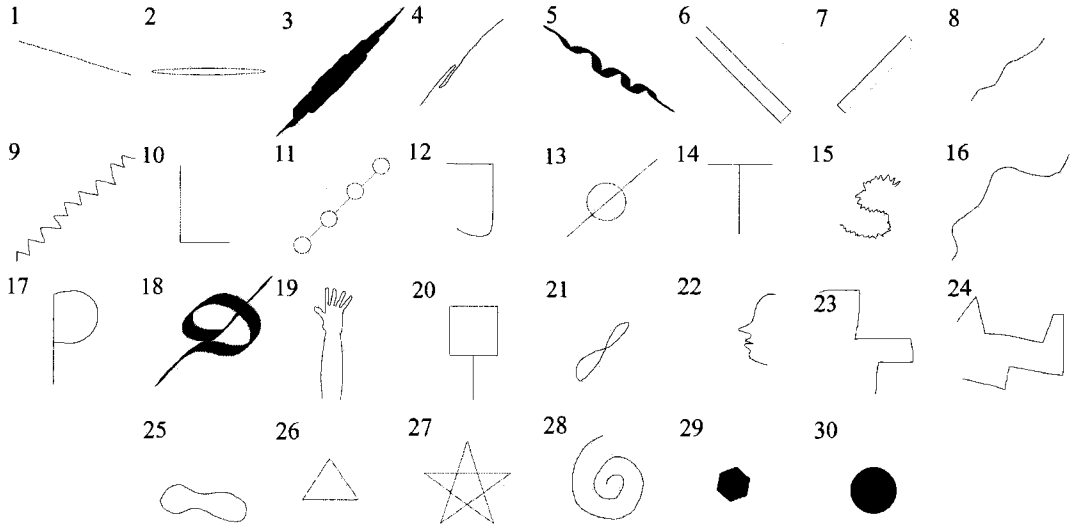


Figure 5.3 Test examples for measuring linearity

The basic framework of each algorithm involved extracting the black pixels from the image and putting them into an 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. 1 means perfectly linear, whereas lesser values mean the shapes are less and less linear.

The table outlining the results of all six algorithms is shown below. The columns are labelled after the algorithms: Average Orientations (AO), Rotation Correlation and Eccentricity (RC/E), Triangle Heights (TH), Triangle Perimeters (TP), Triplet Smoothness (TS) 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 results of each algorithm were compared to this column when their accuracy was determined. The correlation values of each algorithm to the average human measurements are seen at the bottom of the table.

The comparison to human perception was done by correlating the results of each algorithm with the AHP column. 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 5.1.

Table 5.1 RESULTS OF LINEARITY ALGORITHMS

	AO	RC/E	TH	TP	TS	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:	.859	.803	.856	.833	.849	.809			

We extended this work to measure linearity of an ordered set of points in the next chapter. The proposed measures are adopted by also considering the ordering of points when projected along the orientation line.

Chapter 6 Measuring linearity of ordered point sets

Here, we will describe the measures of linearity of ordered sets and the monotonicity measure, which constitute the contributions of this chapter. The monotonicity measure is combined with linearity measures for unordered sets to achieve usable results.

6.1 Average Sorted Orientations

Average sorted orientations works much the same as its average orientations predecessor (see chapter 5), 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 6.1 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 measure of unordered linearity from the previous chapter.

Figure 6.1 Average Sorted Orientations

Algorithm: *average sorted Orientations:*
Input: *k:* the number of pairs;
array of points: $Points=(X_i, Y_i), 1 \le i \le N$;
Output: *linearity* value;

Array *a, b*; //slopes of each line found in loop
For i=1 **to** k **do** {
Take two random points $P_1(x_1, y_1)$, $P_2(x_2, y_2)$ from *Points*;
Ensure $P_1<P_2$ in sorted array of points along curve;
$a[i]= x_2-x_1$, $b[i]= y_2-y_1$;
Normalization factor $norm=\sqrt{a[i]^2 + b[i]^2}$;
$a[i]= a[i]/norm$; $b[i]=b[i]/norm$;
}

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

$linearity = \sqrt{A^2 + B^2}$;
if ($linearity<0$) **then** $linearity=-linearity$;
Output *linearity*;

6.2 Triangle Sides Ratio

Here, we take k ordered triplets of points, and measure their average collinearity. 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 6.2 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.

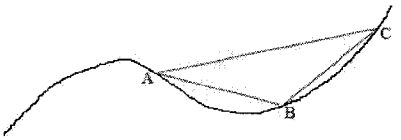


Figure 6.2 Triangle sides ratio example

The minimum linearity value of the algorithm is 0.538, and is obtained for circles. This value was determined experimentally. We stretch out the results by adjusting the linearity as follows:

$linearity = (linearity - 0.538)/0.462$.

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

Algorithm: *triangle sides ratio:*
Input: *k:* the number of triples;
array of points: $Points=(X_i, Y_i)$, $1 \le i \le N$;
Output: *linearity* value;

sum=0; *//sum of all ratios*
For 1 **to** *k* **do** {
Select $P_1(x_1, y_1)$, $P_2(x_2, y_2)$, $P_3(x_3, y_3)$ from *Points* at random;
Ensure that $P_1<P_2<P_3$, in sorted array Points;
Find distances between them: $a=|P_1P_2|$, $b=|P_2P_3|$, $c=|P_1P_3|$;
sum = sum + c/(a+b);
}
linearity=sum/k;
$linearity = (linearity - 0.538)/0.462$;
output *linearity*;

6.3 Monotonicity

Monotonicity measures the behaviour 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. Monotonicity is introduced here to 'correct' the error made by using linearity measures for unordered sets from the previous chapter. For example, tiny (open) ellipses and rectangles are highly linear with these measures, but highly non-monotone, and therefore the product of the two measures is an appropriate measure of linearity for them, if considered as ordered point sets. 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 an orientation line with an angle α which is 90 degrees offset from what is visually the actual orientation line, we repeat the entire procedure for and angle $\alpha+\pi/2$, and select the higher of the two measures. In Figure 6.3, 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.

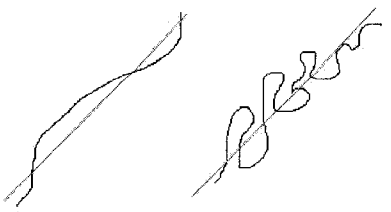


Figure 6.3 Monotonicity examples

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 6.4 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.

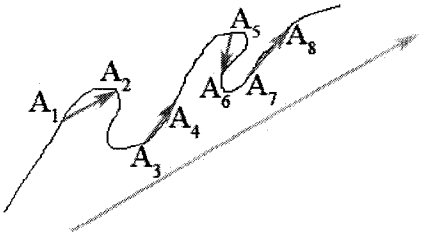


Figure 6.4 Monotonicity functional example

Algorithm: *monotonicity:*
Input: array of points: *Points*=(X_i, Y_i), $1 \le i \le N$;
Output: *linearity* value;

Find center of mass (x_c, y_c) using moments;
Find slope M=tan(α) of orientation of curve
//Vector (1, M) is now the slope of the orientation line

sumMags=0;
sumSigns=0;
For *i*=1 **to** *N-4* **do** {
    Take points $P[i](x_1, y_1)$, $P[i+4](x_2, y_2)$ from *Points*;
    vectorX= x_2-x_1; *vectorY*= y_2-y_1;
    dotProduct= *vectorX*+ *vectorY*M;*

sign=0;
if (*dotProduct>0)* **then** *sign=1* **else** *sign=-1;*
mag= $\sqrt{vectorX^2 + vectorY^2}$
*sumSigns = sumSigns + sign*mag;*
sumMags = sumMags + mag;
}
linearity = sumSigns/ sumMags;
if (*linearity*<0) **then** *linearity=-linearity;*

Repeat entire **for** loop for $\alpha 1=\alpha+\pi/2$, which results in a new linearity value, *linearity1*;

If (*linearity*< *linearity1*) **then** *linearity=linearity1*;
Output *linearity.*

6.4 Experimental data

The linearity algorithms were tested on a set of 25 non trivial shapes, shown in Figure 6.5. These shapes were assembled by hand and are meant to cover a wide variety of non trivial curves. 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 a procedure is called by large curves, the speed may become a concern. We also added speedup procedures to determine the sample size k so that linearity measure has sufficient accuracy. The speed improvement was made using basic probability theory for confidence intervals [HT]. This addition did not change the linearity measures but made the already fast algorithms even faster. Details are given in section 6.5.

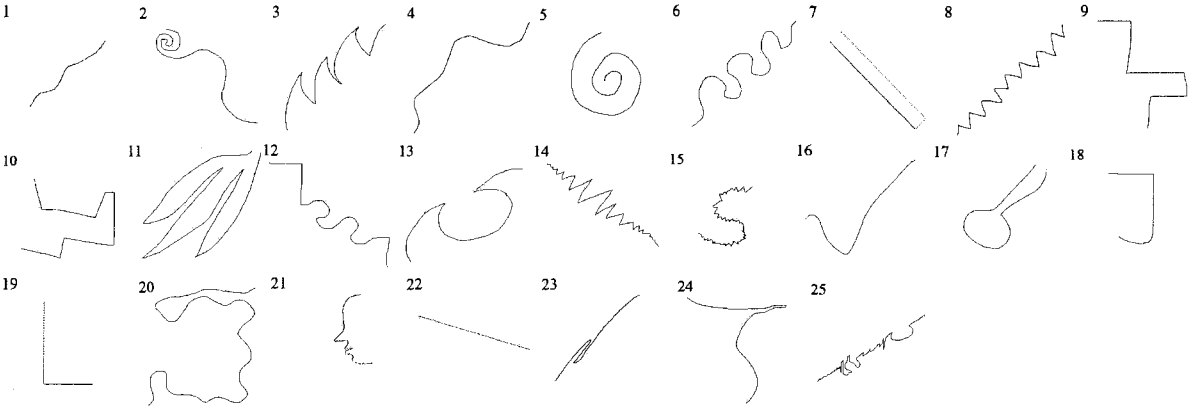


Figure 6.5 Linearity test set

Table 6.1 LINEARITY RESULTS

	AO	EC	TH	TP	CS	EAR	ASO	TSR	MON	AH
1	.95	.99	.85	.97	.80	.93	.98	.99	1.00	.90
2	.35	.56	.29	.37	.23	.47	.71	.71	.65	.47
3	.34	.47	.25	.31	.20	.40	.85	.91	.54	.60
4	.71	.83	.61	.79	.50	.70	.90	.92	.98	.81
5	.01	.02	.00	.00	.00	.02	.27	.18	.16	.06
6	.36	.55	.24	.32	.19	.46	.82	.90	.65	.63
7	.10	.12	.07	.08	.06	.10	.26	.49	.13	.15
8	.83	.96	.73	.85	.66	.88	.94	.95	.98	.82
9	.35	.47	.25	.31	.20	.39	.81	.83	.69	.50
10	.04	.06	.02	.03	.02	.05	.45	.59	.20	.31
11	.05	.09	.02	.03	.01	.08	.47	.25	.16	.10
12	.61	.79	.50	.64	.42	.68	.89	.92	.87	.68
13	.23	.36	.13	.17	.11	.29	.71	.79	.53	.30
14	.57	.71	.44	.53	.38	.63	.91	.95	.75	.80
15	.09	.16	.07	.08	.05	.14	.70	.71	.32	.42
16	.35	.53	.35	.39	.32	.44	.79	.90	.70	.70
17	.05	.07	.03	.04	.02	.05	.25	.26	.09	.10
18	.26	.43	.28	.36	.22	.35	.76	.69	.65	.64
19	.59	.73	.56	.68	.49	.61	.85	.91	1.00	.69
20	.08	.13	.06	.08	.04	.11	.61	.50	.52	.30
21	.44	.62	.42	.54	.34	.52	.86	.82	.80	.57
22	1.00	1.00	.99	1.00	.99	1.00	1.00	1.00	1.00	1.00
23	.64	.66	.56	.61	.53	.63	.85	.96	.67	.82
24	.04	.22	.24	.28	.22	.20	.64	.69	.78	.51
25	.41	.50	.36	.42	.32	.45	.91	.97	.51	.73
	.89	.91	.90	.91	.88	.92	.93	.92		

Figure 6.5 shows the test curves used in the linearity experiments. Table 6.1 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 [SNZ] 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. The Average Human (AH) measurements column represents the average results of 30 test subjects. The correlation of these results with each algorithm is seen at the bottom of the table. The two new algorithms show the closest correlation to the human measurements.

Table 6.2 shows the correlations between pairs of linearity algorithms. We see that the set of unsorted linearity algorithms which were multiplied by the monotonicity factor are highly correlated amongst themselves. The two linearity algorithms which deal directly with ordered sets are also highly correlated with each other. The correlation values between the adapted linearity algorithms and the two new ones are not as high.

Table 6.2 CORRELATION VALUES

	AO	EC	TH	TP	CS	EAR	ASO	TSR
AO	X							
EC	0.98	X						
TH	0.98	0.95	X					
TP	0.98	0.97	0.99	X				
CS	0.96	0.93	0.99	0.97	X			
EAR	0.99	0.99	0.97	0.98	0.95	X		
ASO	0.82	0.88	0.80	0.83	0.77	0.87	X	
TSR	0.79	0.85	0.76	0.79	0.74	0.83	0.93	X

We have covered a variety of methods of assigning linearity values to ordered point sets. The methods that were built to specifically attack this problem: ASO and TSR, proved themselves to be the best according to human measurements.

6.5 Determining sample size

We will discuss now how to determine sample size k so that linearity measure has sufficient accuracy. A parallel goal is to optimize the speed of the algorithm. Let λ be one of considered linearity measures for a set of points, l_i measured linearity value for i-th sample, $1 \le i \le k$, and let χ and s be measured average and variance after taking k samples. The 95% confidence interval for l is $(\chi - 1.96s/\sqrt{k}\ ,\ \chi + 1.96s/\sqrt{k}\ )$ [HT], where $\chi = \sum_{i=1}^{k} l_i$ and $s^2 = \frac{1}{k-1}\sum_{i=1}^{k} (l_i - \chi)^2 = (A - k\chi^2)/(k-1)$, $A = \sum_{i=1}^{k} l_i^2$. When a new $(k+1)$-th sample is taken with average l_{k+1}, the new χ and s are updated (in constant time) via the previous average χ as follows: $\chi \leftarrow (k\chi + l_{k+1})/(k+1)$, $A \leftarrow A + l_{k+1}^2$, $s^2 \leftarrow (A - (k+1)\chi^2)/k$. Suppose that the requested absolute error (with 95% confidence) for χ is t. Therefore $1.96s/\sqrt{k} < t$, and the criterion for stopping further sample selection is $k > 1.96^2 s^2/t^2$. In our experiments, we selected t=0.01.

In application, it may be more common to make a judgment on whether or not the set is linear. Let χ and σ be measured average and standard deviation after taking k samples, $\sigma = (A - k\chi^2)/k$, $A = \sum_{i=1}^{k} l_i^2$. Let U be the threshold linearity measure for accepting a set as being linear. We used U=0.95 in our experiments. The probability that the linearity measure λ is $\le U$ is $\Phi((U-\chi)\sqrt{k}/\sigma)$ [HT], where Φ is normal distribution. The error does not exceed α if $(U-\chi)\sqrt{k}/\sigma \ge z_\alpha$. For α=0.01 we have z_α=2.326 [HT]. Therefore the condition for rejecting a set for being linear is $\sqrt{k} \ge z_\alpha \sigma/(U-\chi)$. The set can be accepted as linear if it was not rejected and the measured value is $>U$ with sufficient accuracy, as elaborated above.

Chapter 7 Related properties of linearity and elongation

The elongation of an object is understood to be something that gives an idea of the length vs. the width of that object. The Webster's dictionary definition of this term articulates that elongation is 'the quality of being elongated'. A further search of the term 'elongated' gives 'having notably more length than width'. This is hardly a concise definition, so we present our own set of definitions to accurately define the term 'elongation'. Measuring elongation of a finite set of points in 2d space is equivalent to measuring the linearity of the same set. The linearity of a point set indicates how close this set is to a straight line. We have compared methods measuring linearity with methods of measuring elongation in literature, against the same test set. By correlating the results of both approaches, we were able to empirically show a strong correlation between the two ways of measuring what appears to be the same thing. Elongation/linearity is a useful tool in shape classification tasks in image processing, which is why we devote this article to studying it further.

We consider 5 methods of finding linearity and 5 methods of finding elongation. The linearity algorithms are the ones we discuss in chapter 5. Three of the elongation measures are also taken from chapter 5 and one from [W]. The fifth measure of elongation is the standard area based method proposed in [H1]. The 'eccentricity' measure from [W] was actually at one point used as a linearity measure in chapter 5. Here it is once again considered an elongation measure. The algorithms are sensitive to large extrusions in the curve but they mainly do not react to small ones which could be due to noise.

There are many publications that deal with elongation: [H1, JKS, KR]. The standard measure of shape elongation is derived from the definition of shape orientation that is based on the axis of the least second moment of inertia. Precisely, the axis of the least second moment of inertia ([H, JKS, KR]) is the line which minimizes the sum of the squares of distances of the points (belonging to the shape) to the line.

Here, we will describe the test methodology of the linearity and elongation measures and present the test sets of shapes. The test sets are shown below. Figure 7.1 shows the set of closed shape contours, most of which were taken from the test sets used in [R3]. Figure 7.2 shows the area based curves test set. Figure 7.3 shows the open shape contours test set. There are 23 closed shape contours, 25 area based shapes and 25 open shape contours. Each contour based shape has between 300 and 800 pixels. The area based shapes have roughly the same number of boundary pixels, but since their areas are also considered, their pixel count is significantly higher.

It was discovered that all of the algorithms for linearity and elongation can be applied to each test set. The standard elongation measure is area based and thought to be only applicable to area based shapes. However, we have shown that even the standard MMR measure can be applied to boundary based shapes, by applying the same formula on just the boundary pixels rather than all of the pixels inside a shape. The new measure will be referred to as Standard Boundary (SB).

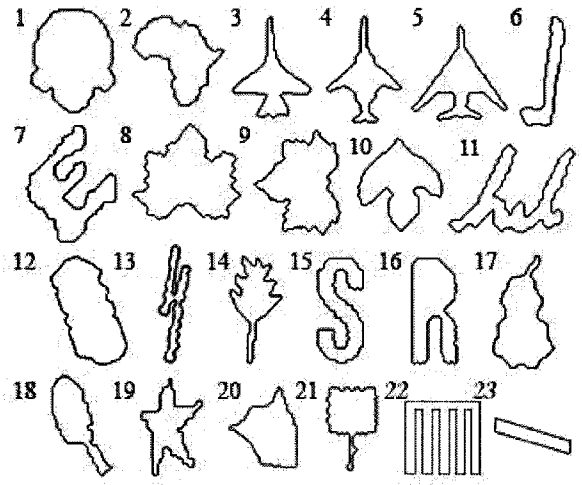


Figure 7.1 Closed shape contours

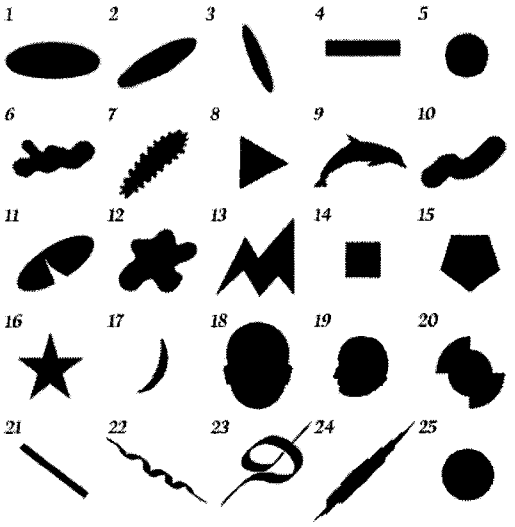


Figure 7.2 Area based shapes

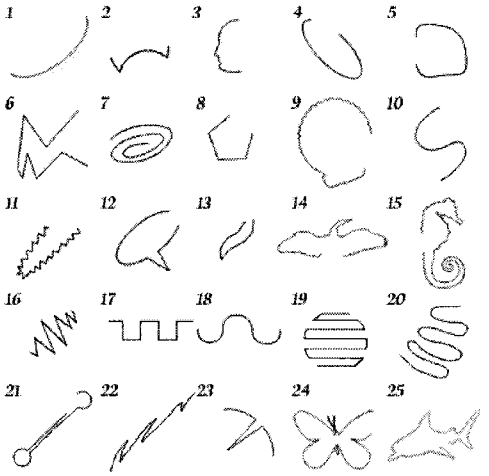


Figure 7.3 Open shape contours

7.1 Experimental Data and Results

Table 7.1 shows the linearity and elongation results for the closed shape contours seen in Figure 7.1.

Table 7.1 LINEARITY AND ELONGATION FOR CLOSED SHAPE CONTOURS

	AO	TH	TP	CS	EAR	E	SB	SZ	ZS
1	.14	.13	.16	.11	.14	.14	.26	.18	.15
2	.27	.13	.14	.12	.25	.28	.43	.06	.05
3	.23	.17	.24	.13	.38	.45	.62	.14	.12
4	.31	.19	.25	.16	.46	.55	.71	.27	.22
5	.06	.08	.10	.06	.14	.15	.27	.08	.08
6	.71	.53	.65	.45	.76	.89	.94	.64	.60
7	.16	.01	.01	.01	.24	.26	.42	.44	.48
8	.11	.07	.07	.06	.13	.14	.25	.04	.04
9	.07	.05	.06	.03	.23	.26	.41	.22	.18
10	.09	.03	.05	.02	.05	.05	.12	.27	.22

11	.37	.12	.17	.10	.34	.40	.57	.56	.58
12	.33	.16	.22	.12	.41	.49	.66	.32	.34
13	.65	.41	.52	.34	.78	.91	.95	.68	.64
14	.41	.29	.37	.23	.55	.66	.80	.41	.36
15	.42	.18	.25	.14	.54	.64	.78	.34	.36
16	.44	.21	.29	.16	.47	.56	.72	.62	.56
17	.44	.20	.26	.16	.46	.55	.71	.36	.32
18	.57	.38	.51	.30	.62	.75	.86	.51	.53
19	.30	.10	.13	.09	.40	.47	.64	.39	.36
20	.12	.01	.00	.01	.16	.17	.29	.22	.21
21	.31	.16	.23	.12	.35	.41	.58	.29	.26
22	.08	.04	.01	.04	.01	.01	.01	.78	.78
23	.73	.66	.79	.59	.87	.97	.98	.76	.77

Table 7.2 shows the linearity and elongation results for the area based shapes seen in Figure 7.2. Table 7.3 shows the linearity and elongation results for the open shape contours seen in Figure 7.3. The first 5 columns of each of the three tables list the linearity results of each shape. They are: Average Orientations (AO), Triangle Heights (TH), Triangle Perimeters (TP), Contour Smoothness (CS), and Ellipse Axis Ratio (EAR). The Elongation measures are listed in the last four columns, and they are: Eccentricity (E), the standard measure of elongation as referred to in [ZKF] (MMR), our new measure *standard boundary* SB which is an elongation measure for just the boundary points of the area shapes as calculated by the same formula as MMR, the first method of elongation defined in chapter 4 (SZ), and finally the convergent method of elongation as defined in chapter 4 (ZS).

Table 7.2 LINEARITY AND ELONGATION FOR AREA BASED SHAPES

	AO	TH	TP	CS	EAR	E	MMR	SB	SZ	ZS
1	.39	.16	.21	.13	.51	.62	.85	.76	.68	.62
2	.45	.28	.35	.25	.65	.78	.92	.88	.74	.76
3	.60	.37	.46	.31	.72	.86	.96	.92	.77	.78
4	.63	.38	.47	.32	.72	.86	.96	.92	.80	.80
5	.08	.15	.19	.12	.00	.00	.00	.00	.01	.00
6	.45	.19	.24	.16	.54	.65	.87	.79	.41	.36
7	.54	.29	.37	.25	.64	.77	.92	.87	.19	.17
8	.03	.10	.09	.10	.01	.01	.00	.02	.15	.13
9	.40	.34	.44	.26	.55	.66	.85	.79	.40	.36
10	.47	.29	.37	.24	.61	.74	.91	.85	.58	.61
11	.39	.19	.27	.12	.53	.64	.86	.78	.34	.37
12	.19	.10	.10	.10	.26	.29	.50	.45	.28	.29
13	.25	.07	.10	.05	.33	.38	.57	.55	.62	.58
14	.06	.02	.01	.04	.00	.00	.00	.00	.01	.01
15	.09	.10	.14	.06	.01	.01	.00	.01	.05	.04
16	.00	.08	.06	.07	.01	.01	.01	.02	.05	.04
17	.71	.57	.73	.47	.71	.85	.93	.91	.62	.64
18	.14	.08	.10	.05	.13	.14	.37	.24	.17	.15
19	.11	.05	.07	.03	.11	.12	.25	.22	.17	.20
20	.14	.09	.13	.06	.25	.28	.62	.44	.14	.17

21	.79	.69	.80	.63	.87	.97	.99	.98	.84	.86
22	.85	.74	.86	.67	.91	.98	.99	.99	.66	.71
23	.39	.15	.23	.12	.51	.61	.61	.76	.55	.57
24	.84	.63	.78	.55	.87	.97	.98	.98	.91	.93
25	.04	.09	.12	.05	.00	.00	.00	.00	.01	.01

In order to compare our results, we correlate the relevant columns in each table to see if elongation and linearity are related. In each table we see the linearity algorithms listed as the columns and the elongation measures listed as the rows. Each cell represents the correlation value between the measures of the corresponding linearity and elongation for a set of curves.

Table 7.3 LINEARITY AND ELONGATION FOR OPEN SHAPE CONTOURS

	AO	TH	TP	CS	EAR	E	SB	SZ	ZS
1	.85	.78	.93	.70	.80	.92	.96	.59	.65
2	.70	.67	.81	.59	.80	.92	.96	.16	.14
3	.53	.54	.70	.43	.61	.74	.85	.29	.32
4	.50	.33	.40	.27	.57	.69	.82	.64	.70
5	.09	.17	.19	.14	.21	.23	.31	.28	.30
6	.05	.12	.15	.11	.16	.17	.30	.48	.41
7	.38	.13	.21	.09	.40	.46	.64	.67	.65
8	.07	.21	.25	.17	.25	.28	.44	.32	.37
9	.00	.20	.26	.16	.23	.26	.40	.15	.16
10	.19	.15	.19	.11	.44	.52	.68	.50	.54
11	.46	.27	.34	.23	.56	.67	.82	.26	.23
12	.25	.07	.07	.07	.17	.18	.33	.37	.40
13	.53	.29	.39	.23	.67	.81	.89	.74	.79
14	.62	.37	.52	.28	.68	.81	.90	.42	.37
15	.38	.27	.35	.22	.57	.69	.82	.13	.13
16	.39	.21	.30	.16	.53	.64	.77	.71	.76
17	.50	.31	.41	.24	.65	.78	.88	.08	.08
18	.48	.39	.53	.29	.60	.73	.84	.05	.06
19	.04	.05	.03	.05	.07	.07	.16	.88	.86
20	.21	.04	.08	.04	.32	.37	.54	.65	.65
21	.79	.73	.84	.67	.87	.97	.98	.72	.76
22	.84	.73	.84	.66	.90	.98	.99	.86	.89
23	.07	.13	.16	.12	.27	.31	.46	.44	.47
24	.18	.02	.03	.02	.25	.28	.43	.15	.16
25	.45	.22	.29	.18	.56	.67	.80	.34	.36

Table 7.4 shows the correlation values for the area based shapes seen in Figure 7.2. Here we see that the correlation values are all very high in each cell. The AO and EAR methods best correlate to the elongation schemes of E, MMR, and SB. We notice that the MMR and SB methods have nearly identical correlation values with each of the linearity measures in Table 7.4. We further examine the relationship between the MMR elongation measure for area and boundary shapes by correlating their results. It was found that these two measures have a correlation factor of .989. This is strong evidence that the area based measure, and the

moment functions that it relies on can be used on boundary shapes as well. For this reason, the MMR measure was also compared to the open and closed shapes in Figures 7.1 and 7.3.

Table 7.4 CORRELATIONS FOR AREA BASED SHAPES

	AO	TH	TP	CS	EAR
E	0.966	0.826	0.853	0.810	0.998
MMR	0.898	0.728	0.761	0.707	0.961
SB	0.924	0.743	0.777	0.724	0.980

Table 7.5 shows the correlation results of the linearity and elongation measures on open and closed curves from Figures 7.1 and 7.3. We immediately notice that the correlation values are quite high for the Eccentricity and MMR elongation measures compared to all of the linearity measures for both open and closed shapes. The SZ and ZS measures for closed curves are not correlated as highly to the linearity values for closed shapes, and it are not correlated at all with the linearity values for open shapes.

Table 7.5 CORRELATIONS FOR OPEN AND CLOSED CURVES

		AO	TH	TP	CS	EAR
Closed	**E**	0.960	0.898	0.921	0.875	0.998
	SZ	0.638	0.574	0.557	0.584	0.549
	ZS	0.621	0.552	0.533	0.563	0.529
	SB	0.920	0.819	0.854	0.790	0.976
Open	**E**	0.946	0.842	0.874	0.813	0.996
	SZ	0.186	0.097	0.066	0.139	0.098
	ZS	0.199	0.121	0.088	0.162	0.119
	SB	0.907	0.779	0.820	0.742	0.978

The reason for such low correlation values between the linearity measures and the elongation of open shapes as calculated by SZ and ZS lies in the way elongation is calculated in for these measures. Consider Figure 7.4 for clarification. There we see three shapes that would have the same elongation value according to the non convergent definition of elongation defined in chapter 4. This happens since their elongation definition is the ratio of horizontal and vertical edges projected onto the x and y axes. We see that since shapes *a*, *b*, and *c* in Figure 7.4 have the same ratio of horizontal and vertical edges, their elongation value would also be the same. Shapes 4, 5, 6, 7, 9, 10, 11, 16, 17, 18, 19, 20 and 23 in Figure 7.3 all exhibit properties such as those seen in Figure 7.4. Their elongation values as calculated by SZ and ZS are consequently much lower than they should be, and therefore when the set of results is correlated with the linearity measures, no direct link can be seen. This is not a reflection of a non existent link between elongation and linearity, it is a testament to situations where the elongation measure of chapter 4 does not perform adequately.

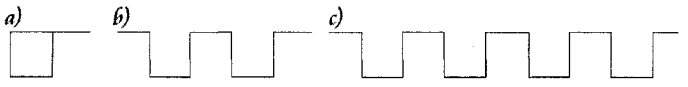


Figure 7.4 Three shapes having the same elongation according to [SZ]

A two tailed, paired t-test of the MMR and SB measures on the area based shape set yields a value of 0.0161. Their mean difference of measures is 0.03096, which means that on average, the measures produce results which vary by 3%. A confidence interval of 95% specifies that the measures will produce values that will differ in the range [0.00627, 0.05565].

Chapter 8 Measuring circularity of point sets

We will present our circularity measures for both ordered and unordered point sets here. The choice of center for each shape is an important factor in measuring circularity. We have used two methods for finding the center of each shape. The first method is the general center of gravity of the shape, which corresponds to a per-coordinate average value of each pixel in the shape. The true center finding method takes the median point value of k triplets sample points belonging to the curve. The results of all tests cases are presented as well.

8.1 Adapting Linearity measures to measure Circularity

The linearity measures presented in chapter 5 were used to measure circularity here. We did not modify the measures themselves. Only the input to each measure was modified. The intended input for the linearity functions was an array of n Cartesian pixel coordinates in the form (x_i, y_i). The Cartesian pixel array was transformed into an array of polar coordinate pixels. The polar coordinate representation of a point p is of the form (r_p, α_p), where r_p is the distance of the point p from the origin, and α_p is the angle p makes with respect to the x-axis, as seen in Figure 8.1.

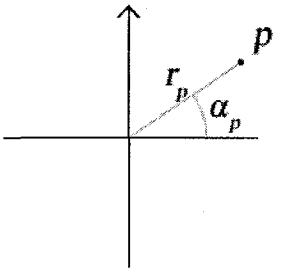


Figure 8.1 Polar coordinate example

Points are transferred from Cartesian coordinates to Polar coordinates as follows: Point (x, y) in Cartesian form would be represented by $\left(\sqrt{x^2 + y^2}, \arctan\ (y/x)\right)$, where $x \neq 0$. This form of point representation is beneficial since circular objects whose points are transferred to polar coordinates appear linear when these points are mapped. Figure 8.2 shows a circle with radius r drawn in planar Cartesian coordinates, and its corresponding polar representation on the right.

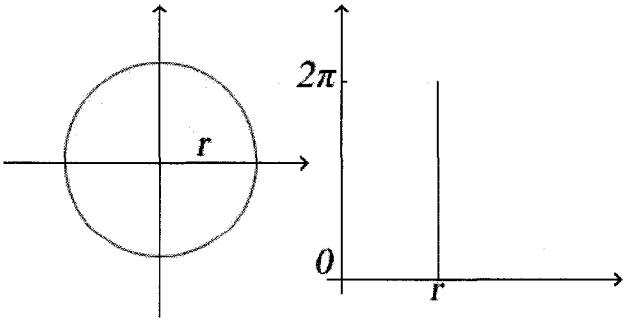


Figure 8.2 Cartesian and polar representations

We can see from Figure 8.2 that measuring the linearity of a polar coordinate point set is synonymous to measuring the circularity of a point set in Cartesian coordinates. The linearity measures from chapter 5 were used to measure the circularity of polar coordinate

input sets. The algorithm that was used to test circularity of a planar set of points with a general linearity method X is presented below.

Algorithm: *Circularity measures:*
Input: array of points: *Points* = (X_i, Y_i), $1 \leq i \leq n$;
Output: *Circularity X, Ordered Circularity MX;*

Begin:
Find center of gravity (X_c, Y_c) of set of points;
Translate the set by $(-X_c, -Y_c)$ so that it is at the origin;
Transform set to Polar Coordinates.
X = linearity value of the polar coordinate set using method X;
M = Monotonicity of point set in polar coordinates;
*XM=X*M;*
Output X, MX;
End.

8.2 Finding the center of a shape

The trivial way of choosing a shape's center is to take the per-coordinate average of all pixels, which is referred to as the center of gravity. This is the method that is usually chosen when measuring any shape property such as linearity or elongation. Choosing the appropriate center of a shape when measuring circularity is more delicate and heavily influences the result of the circularity measure. To illustrate this point, we turn to figure 8.3.

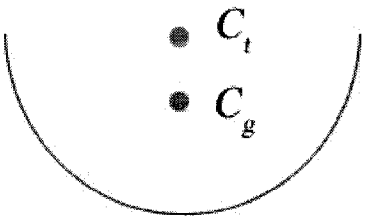


Figure 8.3 Choosing the correct shape center

Here, we see a semi circular shape where the red dot, labeled C_g, represents the center of the shape as determined by the center of gravity method. Transferring the shape to polar coordinates with respect to the red dot would not yield a straight line, but rather a curved one. This would result in a lower circularity measure than expected for the given shape. However, had the green dot been chosen as the center, labeled C_t, the resulting polar coordinate representation would have looked similar to the line seen in Figure 8.2, and a much higher circularity measure would have been awarded.

We have experimented with both types of center finding methods in this work. In order to find the 'true center' of a shape, as opposed to its traditional center of gravity, we sampled k triplets of points from its point set. From each triplet, we found the center (X_{tc}, Y_{tc}) that the points define. Figure 8.4 shows the idea behind finding each 'true center'.

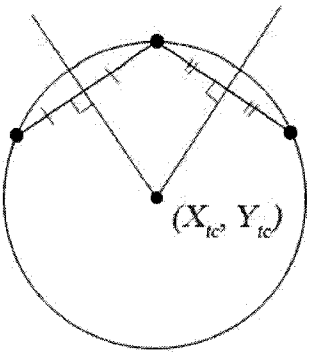


Figure 8.4 Finding the center from 3 points

It is expected that each triplet of points will yield a different center (X_i, Y_i). To choose a unique center for the shape, we individually sort the k center values per coordinate, and choose the median per coordinate value to be the true center, (X_{tc}, Y_{tc}).

8.3 Experimental data

The circularity algorithms were tested on a set of 25 non trivial shapes. These shapes were assembled by hand and are meant to cover a wide variety of non trivial cases. Each shape comprises between 300 and 700 pixels. All shapes are digital, which is why the circularity measures are not ideal.

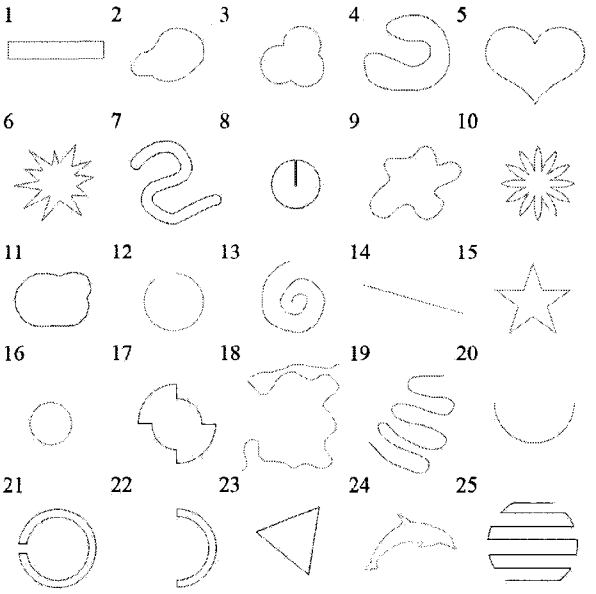


Figure 8.5 Circularity test set

Four sets of measurements are presented here. One set is for the circularity measure where the center of the shape is chosen to be its center of gravity and the points are unordered, the next is the case where the center of gravity is chosen to be the 'true center' and the points are unordered, the third and fourth measures are identical to the first two except that the points of each shape are ordered. In order to get an idea of which measure best measures circularity, we compared them to human measurements of circularity. A group of 20 colleagues volunteered to give their analysis of the shapes in Figure 8.5. Each measure was correlated to the average human results to find the one that best models them. Figure 8.5 shows the test curves used in the circularity experiments.

Table 8.1 holds the circularity values as measured by each method on unordered point sets. The actual measures are in the interval [0, 1], but they are presented as integers in the interval [0, 100] in the tables to save space. The *AOC* column refers to the *Average Orientations* Circularity measure that takes the center of gravity as the center of each shape. *AOT* is the average orientations circularity measure that adopts the 'true center' method when calculating circularity. The *TSC* column refers to the *Triplet Smoothness* Circularity measure that takes the center of gravity as the center of each shape. *TST* is the *Triplet Smoothness* circularity measure that adopts the 'true center' method when calculating circularity. The *P* column shows the results of the measure proposed by [P], and the 'center of gravity' was used as the shape center. The *CKT* column shows the results of the measure proposed by [CKT]. *LS* is the circularity measure proposed by [LS]. The *C* column shows the results of the standard measure of circularity. The perimeter of each shape was calculated as in [KA]. The HM column depicts the average human perception of the circularity of each shape. The correlation values of each measure with human perception are found at the bottom of the table.

We see that some of the entries in the *CKT*, *LS* and *C* columns are empty because these measures are not defined for open shapes. Even when omitting these open curves from consideration, we see that the first four measures more closely follow the average human measurements.

Table 8.1 CIRCULARITY RESULTS FOR UNORDERED POINT SETS

	AOC	AOT	TSC	TST	P	CKT	LS	C	HM
1	51	50	32	32	88	69	30	42	12
2	86	83	61	59	98	91	72	78	77
3	90	89	69	67	99	90	79	76	64
4	74	70	38	36	94	62	53	36	61
5	75	74	46	46	98	85	70	68	45
6	77	71	51	48	98	36	67	13	40
7	74	75	50	50	93	39	18	15	15
8	81	80	69	70	92	77	95	55	97
9	75	74	51	49	98	72	69	50	44
10	76	77	48	48	97	29	64	10	53
11	96	96	81	82	99	96	80	85	85
12	99	99	90	98	100				94
13	68	66	39	36	93				65
14	9	88	30	83	82				0
15	66	66	34	32	96	53	61	26	27
16	100	100	98	99	100	99	98	93	100
17	78	89	72	71	98	82	66	60	38
18	59	66	35	40	98				12
19	60	57	36	29	93				4
20	75	99	61	95	93				62
21	91	89	76	76	99	33	0	12	91
22	76	86	64	77	94	44	12	21	56
23	86	89	70	71	97	79	69	59	20
24	45	44	31	20	90	52	41	25	29
25	75	73	53	48	94				43
cor	0.74	0.58	0.75	0.56	0.52	0.33	0.35	0.42	

Table 8.2 shows the results of the circularity measures as applied to ordered data. The *AOM* column represents the average orientations circularity measure multiplied by its corresponding monotonicity factor. *AOTM* shows the average orientation circularity considering the 'true center' of each shape, multiplied by the corresponding monotonicity. The *TSM* column represents the Triplet Smoothness circularity measure multiplied by its corresponding monotonicity factor. *TSTM* shows the Triplet Smoothness circularity considering the 'true center' of each shape, multiplied by the corresponding monotonicity. Column *PM* shows the *P* circularity measure multiplied by the monotonicity values of each shape. This is the only other measure, apart from our own, that monotonicity was applicable to since *P* is defined for open curves. The other measures are area based, and monotonicity cannot be calculated for them. The monotonicity values, with respect to the center of gravity of each shape, are seen in column *MON*. The *HMM* column shows the average human measurements of circularity considering the monotonicity of shapes.

Table 8.2 CIRCULARITY RESULTS FOR ORDERED POINT SETS

	AOM	AOTM	TSM	TSTM	PM	MON	HMM
1	51	42	32	27	88	1.00	8
2	86	83	61	59	98	1.00	75
3	90	89	69	67	99	1.00	63
4	23	23	12	12	30	0.32	20
5	75	74	46	46	98	1.00	38
6	74	68	49	46	93	0.95	30
7	30	30	20	20	38	0.41	8
8	28	28	24	25	32	0.35	80
9	75	74	51	49	98	1.00	36
10	76	77	48	48	97	1.00	37
11	96	96	81	82	99	1.00	87
12	35	25	32	24	60	0.60	93
13	8	9	5	5	12	0.12	54
14	0	1	0	1	10	0.13	1
15	66	65	34	32	96	1.00	25
16	100	100	98	99	100	1.00	100
17	64	30	59	24	81	0.82	46
18	5	8	3	5	24	0.25	10
19	2	0	1	0	3	0.04	4
20	28	0	22	0	93	1.00	65
21	29	30	24	26	32	0.32	38
22	31	41	26	37	38	0.41	26
23	86	89	70	71	97	1.00	23
24	21	2	14	1	42	0.47	16
25	7	7	5	5	9	0.10	16
cor	0.49	0.43	0.61	0.55	0.44		

Considering both tables seen above, we can say that the circularity of unordered sets is much more easily agreed upon by both humans and computers. One of the most unexpected results that occurs in our measurements is the circularity value of shape 14 when measured by the *AOT* and *TST* methods. The resultant circularity is extremely high (0.88) for a straight line segment. The reason this happens is that the 'true center' for such a line segment is found to be vary far away from the point set. Figure 8.6 illustrates this point.

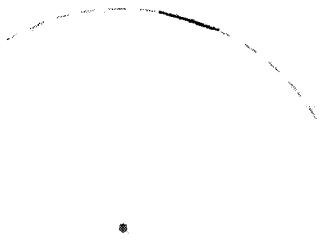


Figure 8.6 True center overfitting

The green center is a point that might be found to be the true center of the dark line in Figure 8.6. This line is highly non-circular, as confirmed by the human measurements, yet the true center assigned to it makes it seem part of a large circle, and hence drastically upgrades its circularity value. In order to avoid such discrepancies, the shape must cover a larger percentage of the overall circle to be considered circular.

The 'true center' method produces circularity results that perform better than the 'center of gravity' method when it comes to semi circular shapes, if the later are to be regarded as fully circular shapes. The new circularity measures that were proposed here outperformed the ones from literature when compared to human measurements.

Overall, the measure that best correlated with human perception on unordered point sets was *TSC*. For ordered sets, *TSM* proved most compatible. Aside from our own proposed method, the *P* measure was the closest competitor from the literature.

Here we investigated the circularity of point sets, or how close these points are to a circle, when considered together. A related problem is to measure how close a set of pixels is to a digital circle. Digital circles are sets of pixels that represent circles in digital space. This problem is studied in [KA, CKT, SNS, LWL, and B1].

Recently, there have been several applications that concentrate on finding circular objects in images. They range from identifying onions in gardens, to finding tires on cars when seen from the side. Future applications might include identifying individual cells based on their boundaries in medical imaging.

Chapter 9 Measuring Ellipticity of point sets

Our new ellipse measures are presented here in two sections. The first deals with fitting an ellipse to point data, and the second measures the ellipticity of the point set by rating the accuracy of the ellipse fit. The quality of the fit relies on an accurate method of finding the center of the shape. We propose two ways of finding shape center. The standard way is by considering the center of gravity, and the other is by finding the true center of the shape.

9.1 Fitting an ellipse to a set of points

Here we describe the algorithm that fits an ellipse to a set of points. Its input is just the set of points, and it outputs the locations of the optimal foci locations, along with the major and minor axes of the fit ellipse and the angle of orientation of the major axis of the fit ellipse.

We begin by finding the angle of orientation α of the point set via moments. The moment based algorithm sometimes produces orientation angles that are normal to the actual shape orientation. To verify the correctness of the obtained value α, the linearity of the set was measured twice. The first measure was made considering the orientation angle was α, and the second was made considering orientation angle $\alpha + 90^{\circ}$. Linearity can be measured using any one of the linearity measures from chapter 5. The higher of the two linearity values corresponds to the actual orientation of the shape.

The orientation line passes through the selected center and has slope α. We then project all points onto the orientation line, resulting in a new array. The two extremity points *min* and *max* along the orientation line of the new array are found. In Figure 9.1 we see the blue orientation line which is also the line on which the points on the shape are projected. G is the center of the shape. Foci f_1 and f_2 will be determined by the foci finding procedure to follow.

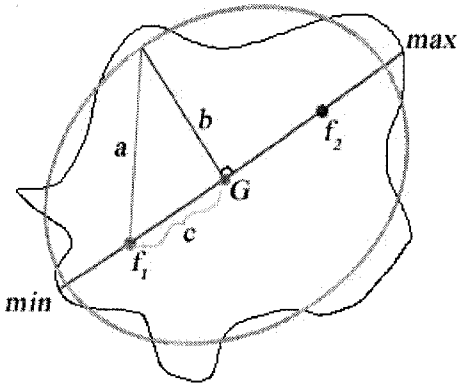


Figure 9.1 Orientation line with foci, *min*, *max*, *a*, *b*, *c*, and *G*

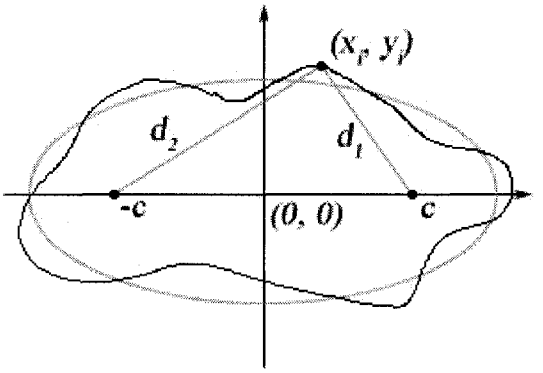


Figure 9.2 Varience of summed foci distances

9.1.1 Finding optimal foci for ellipse fitting

For simplicity, we assume that the point set has been translated such that its center is at the origin. Also, the point set has been rotated such that its orientation line lies on the x axis. In Figure 9.2, the distances to the foci are:

$$d_1 = \sqrt{(x_i - c)^2 + y_i^2}\ , \text{ and } d_2 = \sqrt{(x_i + c)^2 + y_i^2}\ .$$

Therefore, we have

$$D_i = d_1 + d_2 = \sqrt{(x_i - c)^2 + y_i^2} + \sqrt{(x_i + c)^2 + y_i^2}$$

We need to find c for which the values D_i have the smallest possible variance. Variance is defined by:

$$f(c)=(N-1)\sigma^2=\sum_{i=1}^{N}D_i^{\,2}-\frac{1}{N}\left(\sum_{i=1}^{N}D_i\right)^2.$$

This is a continuous function and thus has a minimum value c such that $f'(c)=0$.

$$f(c)=\sum_{i=1}^{N}\left(\sqrt{(x_i-c)^2+y_i^{\,2}}+\sqrt{(x_i+c)^2+y_i^{\,2}}\right)^2-\frac{1}{N}\left(\sum_{i=1}^{N}\left(\sqrt{(x_i-c)^2+y_i^{\,2}}+\sqrt{(x_i+c)^2+y_i^{\,2}}\right)\right)^2$$

$$\Rightarrow f'(c)=2\sum_{i=1}^{N}\left(\sqrt{(x_i-c)^2+y_i^{\,2}}+\sqrt{(x_i+c)^2+y_i^{\,2}}\right)\cdot\left(\frac{-(x_i-c)}{\sqrt{(x_i-c)^2+y_i^{\,2}}}+\frac{(x_i+c)}{\sqrt{(x_i+c)^2+y_i^{\,2}}}\right)-$$

$$\frac{2}{N}\left(\sum_{i=1}^{N}\left(\sqrt{(x_i-c)^2+y_i^{\,2}}+\sqrt{(x_i+c)^2+y_i^{\,2}}\right)\right)\cdot\sum_{i=1}^{N}\left(\frac{-(x_i-c)}{\sqrt{(x_i-c)^2+y_i^{\,2}}}+\frac{(x_i+c)}{\sqrt{(x_i+c)^2+y_i^{\,2}}}\right).$$

This is a continuous function of one variable, which can be solved by a standard equation root finding technique of numerical analysis, such as the bisection method. Note that, *f(c)* may have local minima and thus multiple solutions for equation *f'(c)*, some of which could even correspond to local maxima. Therefore solving the problem in this direction is not straight forward. We therefore opted for a simple approximate solution that corresponds to a linear search with pixel unit distance steps.

Foci in our implementation are found by inspecting each possible location for the pair between *min* and *G* for f_1, and *G* and *max* for f_2 simultaneously so that $|Gf_1|=|Gf_2|$. For each candidate location foci pair, the distances from each point on the shape to both foci are stored in an array. The variance of this array is calculated for each candidate pair, and the one with the lowest variance corresponds to the best choice for the locations of the ellipse fit foci.

Now that foci are found, we need the median of the sum of distances from each point on the shape to both foci in order to find the length of the major axis a, which equals half of this sum. The distance c from focus f_1 to center G is used to find the length of the minor axis $b=\sqrt{a^2-c^2}$. We now have all of the necessary components of the ellipse fit to be able to evaluate it.

9.2 Assessing the fit quality: minimal variance of summed foci distances

Once the foci of the ellipse fit have been determined, the quality of the ellipse fit can be assessed. This was done by first transforming the original point set into polar representation. This transformation is seen in Figure 9.3.

In Figure 9.3, we see the inherent ellipse property that each point on the ellipse is equidistant to the sum of distances from both foci. We exploit this property when transforming the point set to polar coordinate form. As seen in the bottom part of Figure 9.3, the polar distance value for each point x from the center G will be the sum of distances from x to both foci, $r=d_1+d_2$. The angle α that vector Gx' forms with the x-axis will remain the same as the angle Gx formed with the x-axis. For a perfect ellipse, the resulting shape can be drawn as a circle, but if its polar coordinate shapes are plotted as Cartesian, they would look highly linear, as seen in Figure 4.

Applying a linearity measure to this polar representation results in a linearity value for the modified set of points. This linearity value represents the ellipticity value of the original set.

A normalization was applied to the polar coordinate transformation of points since the angles the points form in the new representation is limited to the interval [0, 360], whereas the length of the radii they form is unbounded. This can result in polar representations that are not proportional to the original shape. To normalize the polar representation, the following operations are performed. The normalization factor $norm = 360 \cdot \sum (r_i - r_{av})^2 / (4a^2)$, where a is the major axis of the ellipse fit, and r_{av} is the average size of the radii. The size of each normalized radius $r_i' = norm * (r_i - r_{min})/(r_{max} - r_{min})$, where r_{min} and r_{max} are the smallest and largest radii in the set. The result is a number that fits a radius R into the interval [0, $norm$].

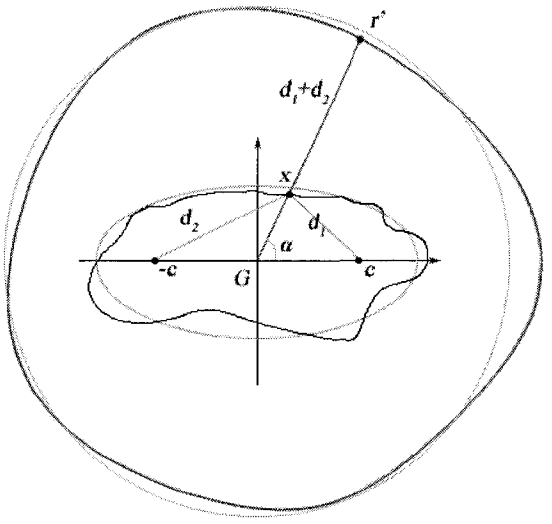


Figure 9.3 Transformating the input set to polar representation

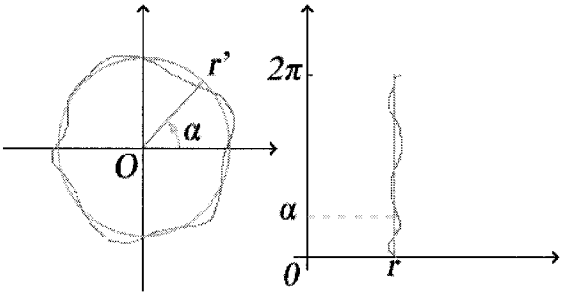


Figure 9.4 Polar point set on a planar graph

This normalization factor represents the size of the interval each radius would be fit into. The larger the variance of the radii, the less the points fit to the ellipse. Therefore the interval they are placed in is larger to make the linearity value of this set smaller.

9.3 Average distance ratio to ellipse and the center

We propose a shape based ellipticity measure that can be applied to open and closed shapes, as shown in Figure 9.5. Let O be the center of fitted ellipse, and V be a point from the original set. Let U be the intersection of line VO with the fitted ellipse that is closer to V, therefore $|VU|<|VO|$. This intersection can be found by using the equations of a line and ellipse, which leads to a quadratic equation. The error measure used is the average of the distance ration to the ellipse and to the center, that is, $1-|VU|/|VO|$. It always returns a number between 0 and 1.

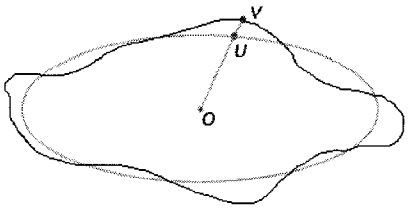


Figure 9.5 Finding the intersect U on line VO

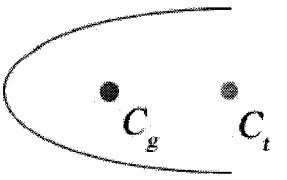


Figure 9.6 Choosing the correct shape center

9.4 Finding the center of a shape

The trivial way of choosing a shape's center is to take the per-coordinate average of all pixels, which is the center of gravity. This is the method that is usually chosen when measuring any shape property such as linearity, orientability or elongation. Choosing the appropriate center of a shape when measuring ellipticity is more delicate and heavily influences the result of the ellipticity measure. To illustrate this point, we turn to figure 9.6.

Here, we see a semi elliptical shape where the red dot represents the center of the shape as determined by the traditional method. Transferring the shape to polar coordinates with respect to the red dot would not yield a straight line, but rather a curved one. This would result in a much lower ellipticity measure than expected for the given shape. However, had the green dot been chosen as the center, the resulting polar coordinate representation would have looked similar to the line seen in Figure 4, and a much higher ellipticity measure would have been awarded.

We have experimented with both types of center finding methods in this work. In order to find the 'true center' of a shape, as opposed to its traditional center of gravity, we sampled k quintuplets of points from its point set. From each quintuplet, we found the center (X_{tc}, Y_{tc}) that the points define via the method proposed in [R6].

9.5 Experimental data

The ellipticity algorithms were tested on a set of 20 closed shapes, shown in Figure 9.7, and 15 open shapes shown in Figure 9.8. These shapes were assembled by hand and are meant to cover a wide variety of non trivial curves. Each of them is comprised of between 100 and 500 points. This number of pixels is common in extracted edges of some computer vision systems.

The center of gravity was close to the true center in almost all of the closed test curves seen in Figure 7. There are however big differences in the ellipticity values of the shapes seen in Figure 7 because the two centers were generally not close.

Table 9.1 holds the ellipticity values for closed curves. The three algorithms for ellipse fitting found in table 9.2 are the new foci variance fitting measure proposed here, the fitting measure proposed by Voss and Sube [VS], and Rosin's sampling LMedS technique [R6]. This set of three ellipse fitting techniques was evaluated by two separate ellipticity measures: the ellipticity via linearity measure proposed here (average orientation linearity measure was applied in polar space), and the area based measure proposed by Koprnicky, Ahmed, Kamel [KAK]. The right most column in table 9.1 shows the ellipticity values of the E measure in [R3], which does not rely on an ellipse fit of a shape.

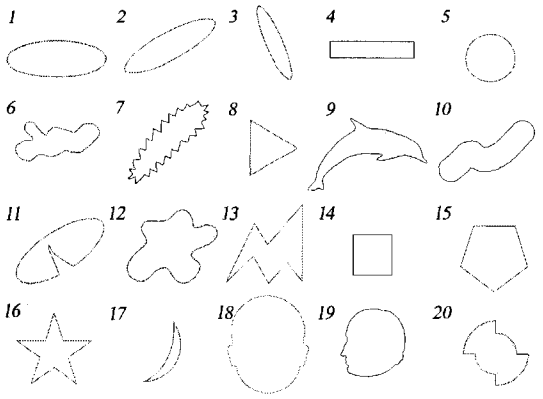


Figure 9.7 Closed test curves

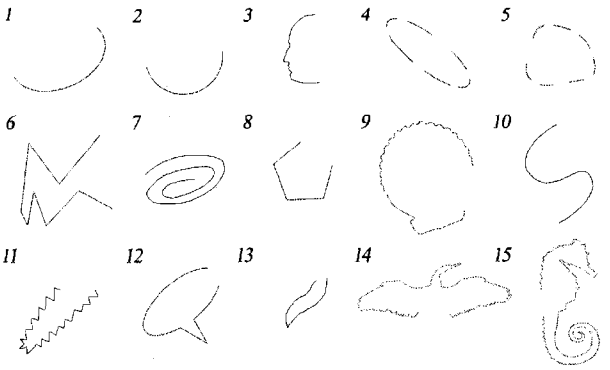


Figure 9.8 Open test curves

The VS [VS] fit algorithm is area based, so to be able to compare it to the shape based ones, the fit it produced based on the area pixels of a shape was evaluated for ellipticity against the perimeter pixels of the same shape. The perimeter pixels were extracted using a 3x3 mask that checked to see if a pixel had at least one neighbour which is not included in the shape. If it had at least one, it was considered as part of the perimeter.

Most of the algorithms agree upon the ellipticity results of the figures which are highly elliptic to the naked eye, as seen in table 9.1. Shape 8 in figure 9.7 is the first major disagreement between the ellipticity fits and measures. The left two shapes of Figure 9.7 show the ellipse fit of our algorithm, and the general fit of the rest of the algorithms for this triangular shape. Our fit is more elongated, and closely follows the triangle on the left hand side. The ellipticity values for this shape vary considerably between measures. Our measure produces the lowest ellipticity value for this shape and we feel that such a low score is merited given that a triangle should not be closely associated with an ellipse.

Shape 16 in Figure 9.7 shares a similar discrepancy in awarded ellipticity scores. The star shape has similar disagreements in ellipticity scores to the triangular shape, and the explanation for these disagreements is also similar. The difference in fits can be seen in the right part of Figure 9.9. The rest of the closed shapes have ellipse fits and values that very closely follow the original shapes.

Table 9.1 ELLIPTICITY RESULTS FOR CLOSED CURVES

	Ellipticity Via linearity			**KAK**			
	Foci var gc	***[VS] VS fit***	***[R6] LMedS***	***Foci var gc***	***[VS] VS fit***	***[R6] LMedS***	***E [R3]***
1	1.00	0.98	1.00	0.95	0.99	0.96	1.00
2	1.00	0.97	1.00	0.94	0.99	0.94	1.00
3	1.00	0.98	1.00	0.96	0.99	0.97	1.00
4	0.92	0.83	1.00	0.82	0.92	0.68	0.91
5	1.00	0.98	1.00	0.98	1.00	0.98	1.00
6	0.83	0.73	0.75	0.68	0.9	0.64	0.72
7	0.99	0.84	0.97	0.8	0.94	0.93	0.9
8	0.27	0.68	0.83	0.59	0.88	0.31	0.68
9	0.51	0.54	0.49	0.44	0.87	0.73	0.35
10	0.84	0.77	0.92	0.77	0.91	0.86	0.76
11	0.99	0.89	0.99	0.84	0.96	0.97	0.87

12	0.75	0.73	0.79	0.69	0.89	0.93	0.78
13	0.77	0.63	0.69	0.24	0.87	0.64	0.55
14	0.4	0.63	0.14	0.34	0.93	0.99	0.91
15	0.85	0.77	0.82	0.89	0.95	0.15	0.97
16	0.45	0.6	NA	0.49	0.86	NA	0.54
17	0.89	0.37	NA	0.45	0.87	NA	0.38
18	0.99	0.45	0.99	0.95	0.44	0.94	0.99
19	0.89	0.32	0.85	0.92	0.33	0.93	0.98
20	0.91	0.39	0.74	0.55	0.94	0.95	0.92

Table 9.2 ELLIPTICITY RESULTS FOR OPEN CURVES

	Ellipticity Via polar linearity			**Distance ratio to ellipse and center**		
	Foci var tc	***Foci var cg***	***[R6] LMedS***	***Foci var tc***	***Foci var cg***	***[R6] LMedS***
1	0.94	0.54	0.88	0.97	0.83	0.86
2	0.98	0.66	0.93	0.98	0.82	0.96
3	0.61	0.39	0.40	0.90	0.82	0.88
4	1.00	1.00	1.00	0.97	0.97	0.98
5	0.96	0.98	0.99	0.72	0.75	0.66
6	0.10	0.71	NA	0.74	0.69	NA
7	0.82	0.83	0.80	0.74	0.76	0.66
8	0.82	0.56	0.71	0.92	0.89	0.79
9	0.91	0.62	0.91	0.95	0.92	0.96
10	0.96	0.22	0.97	0.89	0.79	0.78
11	0.93	0.92	0.96	0.92	0.82	0.88
12	0.79	0.92	0.87	0.86	0.83	0.87
13	0.96	0.98	1.00	0.84	0.86	0.66
14	0.92	0.93	0.87	0.77	0.83	0.78
15	0.61	0.69	0.95	0.71	0.72	0.56

Table 9.2 shows the results of the ellipticity measures as applied to the open curves in Figure 9.8. Our ellipse fits with true and gravity centers based on minimizing variance of summed distances to foci is compared with sampling LMedS method [R6]. Fit accuracies are measured by our two newly proposed methods, polar linearity and average distance ratio to the fitted ellipse and its center.

The choice of center significantly impacts the fit and ellipticity values of the shapes tested here. Shapes 1 and 2 in Figure 9.7 best illustrate the impact of center selection. In Figure 9.10, we see these two shapes coupled with their ellipse fits with both the center of gravity and true center approaches. The light blue dots represent the chosen centers for each shape. The ellipticity values for the shapes in Figure 9.9 are 0.54, 0.94, 0.66, and 0.98 respectively.

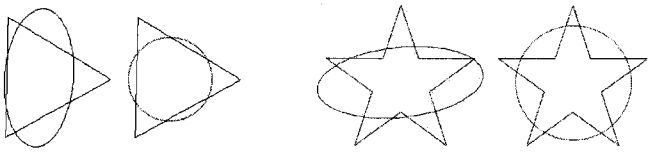


Figure 9.9 Ellipticity of a triangle and a star

Figure 9.10 Center of gravity vs. true center impact

Chapter 10 Measuring Conicity of point sets

Our new conicity measure is presented here. It comprises of fitting a conic to point data, and then, based on the fit, finding a measure of its ellipticity, hyperbolicity or parabolicity. Fitting a conic to a point set is done via Rosin's [R6] 5 point conic fitting algorithm. The evaluation of the fit is done by applying a transformation to the input points (depends on the conic being evaluated: ellipse, hyperbola, or parabola), such that they are transferred to a polar form representation. The linearity of the polar form representation then corresponds to the conicity for the particular shape of the original points.

10.1 Evaluating the conic fit

Once the parameters of the conic have been established (foci, shape orientation...), we can evaluate how well it fits with respect to the shape. We use the properties of each of the potential conics we are dealing with to transform the original Cartesian coordinate set of points into polar form in order to evaluate the linearity of the polar form. This linearity corresponds to the conicity of the original set. If the conic fit was an ellipse then the method proposed in chapter 9 is applied to measure the ellipticity which is also assigned as the shape's conicity value. If the best fit was a hyperbola or a parabola then the corresponding novel methods of measuring these properties are used to measure conicity, as explained in the rest of this section.

10.2 Evaluating hyperbolicity

Hyperbolicity is evaluated in nearly the same way as ellipticity. One of the defining well known properties of a hyperbola is that the difference of distances from each point on the shape to its foci is constant. Figure 10.1 illustrates this point.

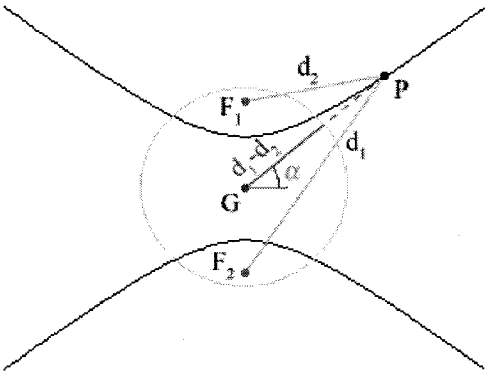


Figure 10.1 Measuring the hyperbolicity of the point set

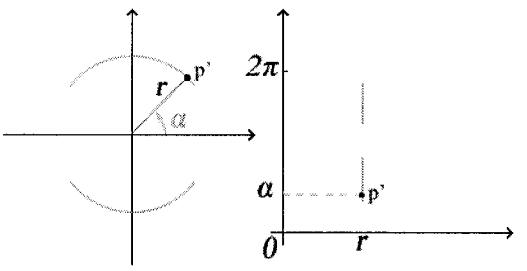


Figure 10.2 Poar point set of a hyperbola on a planar grapb

In Figure 10.1, we see the inherent hyperbola property. We exploit it when transforming the point set to polar coordinate form. The polar distance value for each point P from the center G will be the difference of distances from P to both foci, $r=d_1 - d_2$. The angle α in the polar representation will remain the same as the angle α between GP and the x-axis. For a perfect hyperbola, the resulting shape can be drawn as an incomplete circle, as seen in Figure 10.2, since the radius values for all possible angles α do not exist. If its polar coordinate points are plotted as Cartesian, they would look highly linear. Applying a linearity measure to this polar representation results in a hyperbolicity value for the modified set of points.

Due to the gap between the two parts of a hyperbola, the shape was split in 2 components in its polar form as well, as seen in Figure 10.2. The linearity of two separate line segments taken together increases as the distance between them increases. This property unnaturally increases the hyperbolicity score of any tested shape. To eliminate this bias, the two parts of the set's polar representation are overlapped and linearity is measured on the overlapped set. To overlap the two parts, the angles of the points for which $d_1 - d_2 < 0$ was increased by a value of π, otherwise the angles of the remaining points were left unchanged. The range of angles present in the polar representation of hyperbolas is relatively small, and as such does not permit an adequate representation of the shape. The angles are therefore stretched to the interval $[0, 2\pi]$ to get a normalized representation of the shape. Each angle α is adjusted to α' by the following calculation: $\alpha' = 360 \cdot (\alpha - \alpha_{min}) / (\alpha_{max} - \alpha_{min})$. Each radius r is normalized to yield r' as follows: $r' = (r \cdot k)/(2a)$, where k is a constant whose best value will be experimentally derived. This radius normalization places the range of possible radii in ranges comparable to the angular ones.

10.3 Evaluating parabolicity

Parabolicity is also evaluated by exploiting one of the inherent properties of parabolas, then converting the point set to polar form while making use of this property, and finally measuring the linearity of the polar representation which in effect corresponds to the parabolicity of the original set. Figure 10.3 demonstrates the exploited parabola property used in our work.

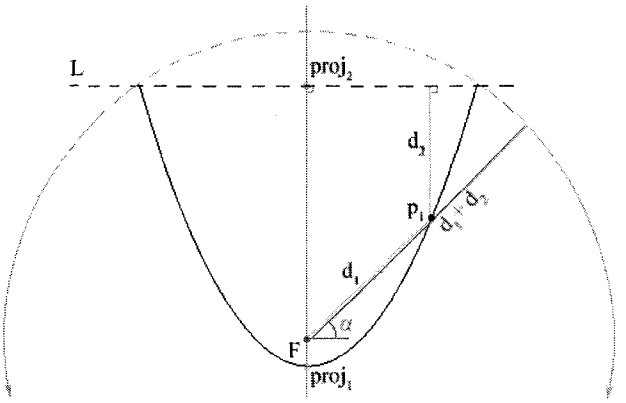


Figure 10.3 Measuring the parabolicity of the point set

Figure 10.3 shows a parabolic curve, along with an arbitrary line L, focus F, and a point on the curve p_1. L is an arbitrary line perpendicular to the axis of symmetry and opposite the focus of the parabola from the vertex. The length of d_1+d_2, where d_1 is the distance from the focus F to the point p_1, and where d_2 is the distance from line L to point p_1, is the same for all points on the shape. This is a similar well known property [PB] to that of the ellipse, except that one focal point is at infinity. Note that an alternative method may compare distances of points from the focus and directrix line, which are supposed to be zero. However this method reduces the circle's radius in polar coordinates to 0, which is unstable and will produce unclear random measures.

To find line L, we first project the original shape points onto the line of symmetry of the parabola. This line of symmetry is the shape orientation line that passes through the focus, as was determined by the 5 coefficients from the point sampling technique described in [R6]. Once the points are projected on the orientation line, the end points $proj_1$, $proj_2$ of this line of projections are found. The end point which is furthest from the focus corresponds to one of

the points on the line L. The slope of line L is the inverse (perpendicular) of the slope of the shape orientation line.

Applying a polar transformation to the set of points such that polar distance value for each point p_i from the focus F will be the sum of distances from p_i to the focus F and from p_i to the line L, $r=d_1 + d_2$. The angle α that vector Fp_1 forms with the x-axis will remain the same in polar coordinates. For a perfect parabola, the resulting shape can be drawn as an incomplete circle since the radius values for all possible angles a do not exist. In Figure 10.3, we observe that the dashed top part of the blue circle would not exist in the polar representation since it does not have corresponding points in its Cartesian representation. If its polar coordinate points are plotted as Cartesian, they would look highly linear. Each radius r is normalized to yield r' as follows: $r'=(r \cdot q)/(|proj_1-proj_2|)$, where the best value for constant q will be experimentally determined, and $proj_1$ and $proj_2$ are the extreme values of the projection point on the line of symmetry of the parabola fit, as seen in Figure 10.3. This radius normalization places the range of possible radii in the interval [0, 360], which corresponds to the interval of possible angles of the radii points. Applying a linearity measure to this polar representation results in a parabolicity value for the modified set of points.

10.4 Experimental data

We implemented Rosin's 5 point sampling method as follows. $K=500$ quintuplets of points were randomly selected from the point set such that no 2 points can be at a distance less than 10 pixels away from each other. This prevents errors in sampling by ensuring that the sample points cover a representative area of the point set.

In practice, our experiments were performed on images where the shapes were represented by pixels with integer coordinates. As such, it is very difficult to obtain a perfect shape, but rather a close approximation. This restriction did not affect the ability of the algorithm to identify ellipses and hyperbolas, but had an impact on parabolas. Since a continuous, perfect parabola is difficult to represent in a discreet coordinate system, such as the one present in images, the condition that specifies that a shape is a parabola: $b^2-4ac=0$, is also difficult to satisfy. The method above gives appropriate results in the cases of ellipses and hyperbolas. Parabolas, on the other hand have only one focus, (where the other one is placed at infinity), which aids us in determining when we are dealing with a parabola, instead of relying on the condition of $b^2-4ac=0$. We determine that we have encountered a parabola when the distance between foci is greater than twice the height of the image the shape is depicted in. This crude approximation of infinite distance between foci serves it purpose in practical terms, although it is not theoretically exact. In case of a parabola, we determine which of the foci is valid by comparing the distance of each focus to the center of gravity of the shape. The one which is closer is kept as the foci of the parabola. Evaluating the quality of the fit was described in chapter 9.

The conicity algorithm was tested on a set of 45 shapes, shown in Figures 10.4, 10.5 and 10.6. These 3 groups of shapes are chosen to equally represent shapes that appear elliptic, hyperbolic and parabolic. These shapes were assembled by hand and each one is comprised of between 100 and 500 points. The angular range used was [0, 360], and the best values for the normalization factor k in the case of ellipses was 180, in the case of hyperbolas it was 30, and q in the case of parabolas was 180 for our figures.

Table 10.1 shows the conicity values for all of the test curves. For each curve, we see the shape designation assigned to it by the algorithm in the 'detected' column, and its

corresponding conicity value in the 'C' column. When these 2 descriptors are taken simultaneously, the algorithm classifies a shape as either an ellipse, hyperbola or parabola, and then assigns a value to this classification which describes how 'elliptic', 'hyperbolic' or 'parabolic' this shape appears to be.

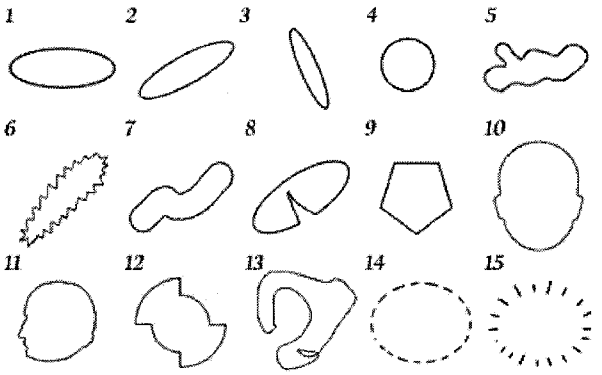


Figure 10.4 Elliptic shapes in the test set

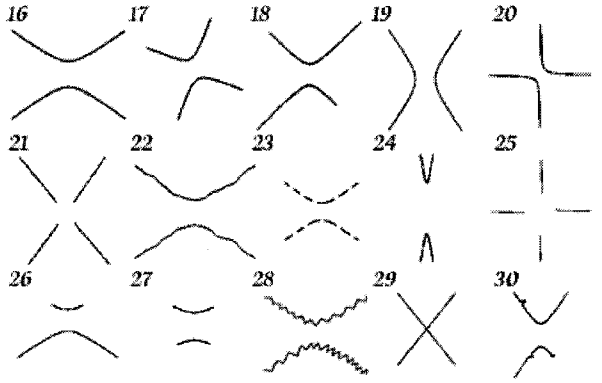


Figure 10.5 Hyperbolic shapes in the test set

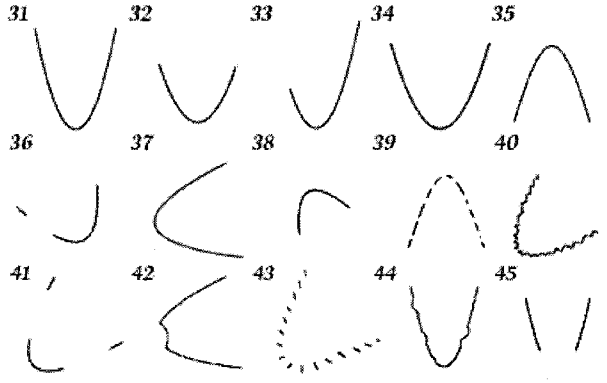


Figure 10.6 Parabolic shapes in the test set

Table 10.1 CONICITY VALUES OF TEST SHAPES

#	detected	C	#	detected	C	#	detected	C
1	ellipse	0.989	16	hyperbola	1.000	31	parabola	0.999
2	ellipse	0.978	17	hyperbola	0.972	32	parabola	0.980
3	ellipse	0.976	18	hyperbola	0.934	33	parabola	0.993
4	ellipse	0.917	19	hyperbola	0.971	34	parabola	0.998
5	ellipse	0.778	20	hyperbola	0.939	35	parabola	0.966
6	ellipse	0.897	21	hyperbola	0.871	36	parabola	0.962
7	ellipse	0.896	22	hyperbola	0.911	37	parabola	0.984
8	ellipse	0.938	23	hyperbola	0.995	38	parabola	0.973
9	ellipse	0.632	24	hyperbola	0.969	39	parabola	0.945
10	ellipse	0.931	25	hyperbola	0.962	40	parabola	0.880
11	ellipse	0.800	26	hyperbola	0.982	41	parabola	0.940
12	ellipse	0.573	27	hyperbola	0.989	42	parabola	0.936
13	ellipse	0.593	28	hyperbola	0.876	43	parabola	0.718
14	ellipse	0.933	29	hyperbola	0.575	44	parabola	0.932
15	ellipse	0.698	30	hyperbola	0.987	45	parabola	0.936

We notice that most of the first fifteen shapes that look elliptic are given relatively high scores and are correctly categorized as ellipses. Intrusions, such as those found in figure 10.4, and missing shape parts, such as those found in shapes 14 and 15 have little effect on the algorithm's ability to correctly classify and measure the conicity of such examples. The conicity result for shape 13 is given the lowest overall elliptic score, since it does not appear to be a very convincing case of an ellipse. It however is even deserving of a place in either the parabola or hyperbola categories of shapes, and was therefore correctly classified as a weak ellipse.

Shapes 16-30 were also correctly classified as hyperbolic by the algorithm. Hyperbolas are inherently comprised of disjoint point sets, but even when these disjoint sets are further broken up into smaller point sets, they can still be identified as hyperbolas, such as in the case of shape 23. Most of these shapes have high hyperbolicity values except for shape 28. This shape looks hyperbolic, and was classified as such, although due to the highly jagged edges it contains, it was given a lower than expected hyperbolicity score.

The algorithm performed well when tested on shapes that appear to be parabolic as well. All fifteen shapes (31 to 45) were correctly classified. Shapes that were missing significant parts (33, 36, 41, 43, 45) were not only correctly classified, but were also given relatively high parabolicity scores.

Chapter 11 Future work

In this thesis, we have proposed measures for linearity, elongation, orientation, circularity, ellipticity, parabolicity, hyperbolicity and monotonicity. We have also investigated orientation of shapes and conic fitting for point sets. We are generally pleased with their performance but there is room for improvement in some of these measures. Applying these measures for solving other problems is a possible direction of future work. This chapter discusses some potential venues for continuing the research presented here.

11.1 Linearity measure and polygonization

We are interested in developing a polygonization algorithm which is based on our linearity measures. The polygonization algorithm should use the linearity measures to divide a curve into visually appealing lines which follow the shape relatively closely, forming a polygon. The curves should be split into appealing line segments based on a combined linearity and distance threshold. The linear polygonization method would iteratively add points to a potential approximating line segment until this threshold is violated. Polygonization starts again from the last point that violated the threshold.

An addition to the simple polygonization method might be a binary search procedure which should produce better results. This is an improvement over the linear search polygonization method. The problem with the first algorithm 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 would also perform faster than the linear extension method. The procedure is illustrated in Figure 8.4, where 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 11.1b), where the new *mid* and *end* points are marked. By continuing the binary search, we find the break point in Figure 11.1c). The segment *Start* to Break *point* is stored as part of the polygonization. The polygonization algorithm will restart from this break point in Fig. 8.4d).

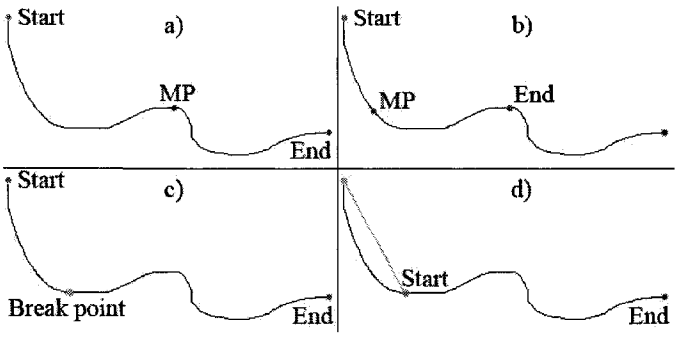


Figure 11.1 Binary polygonization process

11.2 3D and digital linearity

There are a number of possible extensions of our work on measuring linearity. First of all, various applications could be discussed. 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 [BRP, OLC].

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.

It is an interesting problem to measure digital linearity rather than linearity of a set of points. That is, to find how much a set of digital points is a digital line, according to a certain digitization scheme.

11.3 Use of moments on orientation and elongation

We have seen that the measures of elongation and linearity are highly correlated on various sets of data. This can lead us to conclude that these measures are relatively interchangeable, if not completely equivalent. Further experimentation should be done on real world data that actively interchanges these measures in order to be able to better support our assertions.

We have also concluded that the MMR measure can be applied equally to area and shape based figures. This leads us to believe that calculations involving moments of inertia are not strictly limited to area based shapes. A formal proof is missing for this claim, and is left as future work.

11.4 Improving the circularity, ellipticity, and conicity measures

We are generally pleased with our circularity measures since they consistently provide reliable results in various circumstances. The flexibility introduced in choosing a shape center in different ways covers a wide range of input shapes that can now be accurately tested for circularity. Future work for this measure would include finding a way to blend the two center finding methods so that the algorithm automatically knows which to apply in a given case. This would create an automatic and stable measure of circularity in all circumstances.

One point that needs some consideration is normalizing the shape before measuring its linearity in polar form. Currently, the set of points is stretched out to the range $[0, 2\pi]$ in the y-axis of the polar representation. We have been left unsatisfied with the choices of normalization factors for the x-axis of the polar form. Several methods of normalizing the points with respect to the y-axis have been tried, but so far a factor of 1 seems to yield the best results. Further investigation into normalization could make the method more robust and scalable for very large shapes.

The ellipticity measures presented here also cover a wide range of possible inputs, and successfully produce satisfactory results. The [R6] fitting method is more reliable when dealing with shapes that have large sections missing, but becomes unstable in situations where noise is present. Similarly, the foci variance testing method is solid for instances where noise is present and the shape is found in its entirety in the image. It however produces unreliable results when the orientation line cannot accurately be determined, and when large parts of the shape are missing. Combining these two methods would amount to a very good ellipticity measure for a wide range of shapes.

Our work here has demonstrated its capability in identifying shapes that can visually be sorted into one of the three mentioned categories of conics. It however provides inconsistent results when encountering strange shapes that do not easily fit into any category of basic shape. In such instances, due mainly to the sampling technique that was used to fit a conic onto a curve, one of the three conic fits is forced onto a shape that does not appear to be anywhere close to a conic shape, and a low conicity score is awarded. The algorithm has also shown itself to be sensitive when dealing with shapes that appear to be conic, but contain a large number of imperfections or large amount of noise. In such cases, the shape is generally assigned to the correct conic category, but its conicity measure suffers disproportionally to the amount of noise present. It however very accurately detects and fits conics that have large sections missing, but otherwise easily form very good conic shapes.

Measuring hyperbolicity and parabolicity for any shape, including those where the best fit is in the other conic category is left as an open problem. This means finding an acceptable parabola or hyperbola specific fit, regardless of the input data. This appears to be a difficult problem to solve, especially for partially occluded or missing sections of shapes that are otherwise clear examples of parabolas or hyperbolas. The orientation finding techniques used in ellipse fitting cannot be applied to parabolas and hyperbolas that are missing large sections. This would certainly skew the orientation line towards the part of the parabola or hyperbola that is not missing and therefore lead to a fit that cannot overlap the original shape. If the fit is not found properly, then any measure of parabolicity or hyperbolicity cannot yield reliable scores.

11.5 Application of the new methods on real world imagery

Future work in this area includes making the algorithm more robust to noise and subsequently testing it on real world data. Once it is robust enough, it may be included in object detection systems for a variety of quality control applications in manufacturing plans for example. Other uses may encompass using the algorithm to ascertain conicity features in real time computer vision systems. The next step towards this goal is to take all of the measures presented here and combine them into one algorithm that can take any possible input shape and determine what this shape is (line, circle, ellipse, parabola, hyperbola, or none of the above) and give a reliable score as to how confident this analysis is. Then entire scenes can be decomposed into basic shapes and a new field of data mining can use these new tools to analyze them.

12 References

[B] L. Boxer, Computing Deviations from Convexity in Polygons, Pattern Recognition Letters, Vol. 14, pp. 163-167, 1993.

[B1] E. Bribiesca, Measuring 2-D shape compactness using the contact perimeter, Computers & Mathematics with Applications Vol. 33, No. 11, pp. 1-7, 1997.

[B3] F.L. Bookstein, Fitting conic sections to scattered data, Computer Graphics and Image Processing Vol. 9, pp. 56-71, 1979.

[BJM] E. Bribiesca, J.R. Jimenez, V. Medina, R. Valdes, O. Yanez, A voxel-based measure of discrete compactness for brain imaging, Engineering in Medicine and Biology Society, Proceedings of the 25th Annual International Conference of the IEEE Vol. 1, pp. 910–913, 17-21 Sept. 2003.

[Bo] M.J. Bottema, Circularity of objects in images, Proc. IEEE Acoustic, speech and signal processing, ICASSP, Istanbul, 2000.

[BRP] M.A. Badar, S. Raman, P.S. Pulat, Intelligent search-based selection of sample points for straightness and flatness estimation, Journal of Manufacturing Science and Engineering, Vol. 125, pp. 263-271, May 2003.

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

[CAT] J. Cortadellas, J. Amat, F. de la Torre, "Robust Normalization of Silhouettes for Recognition Application," Patt. Rec. Lett., Vol. 25, pp. 591-601, 2004.

[CKT] V. Chatzis, V.G. Kaburlasos, M. Theodorides, An image processing method for particle size and shape estimation, Proceedings of the 2rd International Scientific Conference on Computer Science, Chalkidiki, Greece, 30 September - 2 October 2005.

[D] E.R. Davies, Algorithms for ultra-fast location of ellipses in digital images, Proc. IEE Image Processing and its Applications, Vol. 2, pp. 542 – 546, 1999.

[DD] C. Di Ruberto and A. Dempster, Circularity measures based on mathematical Morphology, Electronics Letters, Vol. 38, No. 20, pp. 1691-1693, September 2000.

[DG] S. Derrode and F. Ghorbel "Shape Analysis and Symmetry Detection in Gray-Level Objects Using the Analytical Fourier-Mellin Representation," Signal Proc., Vol. 84, pp. 25-39, 2004.

[E] N. Efford, Digital image processing, Addison-Wesley, 2000.

[FIT] Set of line fitting functions used to determine orientation, www.mathworks.com/access/helpdesk/help/toolbox/stats/robustfit.html

[FS] H. Freeman, R. Shapira, "Determining the Minimum-Area Encasing Rectangle for an Arbitrary Closed Curve," Comm. of the ACM, Vol. 18, pp. 409-413, 1975.

[FPF] A.W. Fitzgibbon, M. Pilu, R.B. Fisher, Direct least square fitting of ellipses, IEEE Trans. Patt. Anal. And Mach. Intelligence Vol. 21, No. 5, pp. 476-480, 1999.

[H] R.M. Haralick, A measure for circularity of digital figures, IEEE Trans. Syst. Man Cybernet. SMC-4, pp. 394-396, 1974.

[H1] B.K.P. Horn, Robot Vision, MIT Press, Cambridge, MA, 1986.

[HM] V. H. S. Ha, J. M. F. Moura, "Afine-Permutation Invariance of 2-D Shape," IEEE Transanctions on Image Processing, Vol. 14. No. 11, pp. 1687-1700, 2005.

[Ho] B.K.P. Horn, Robot Vision, MIT Press, Cambridge, MA, 1986.

[HT] R.V. Hogg and E.A. Tanis, Probability and Statistical Inference, Prentice Hall, 1997.

[JGX] L. JunWei, J.P. Guillemin, Z. Xiaohui, P. Sarrazin, M. ChenLin, and J.C. Coquille, A study for measuring the sunlit areas of onions and weeds in the field by image processing, Proceedings of ICETS, Beijing, China, 2000.

[JKS] R. Jain, R. Kasturi, B.G. Schunck, Machine Vision, McGraw-Hill, New York, 1995.

[KA] Chul E. Kim, Timothy A. Anderson: Digital Disks and a Digital Compactness Measure, Proceedings of the sixteenth annual ACM symposium on Theory of STOC, pp. 117-124, 1984.

[KAK] M. Koprnicky, M. Ahmed, M. Kamel, Contour description through set operations on dynamic reference shapes, ICIAR, LNCS, Vol. 3211, pp. 400-407, 2004.

[KK] W.Y. Kim, Y.S. Kim, "Robust Rotation Angle Estimator," IEEE Transanctions on Pattern Analysis and Machine Intelligence, Vol. 21. No. 8, pp. 768-773, 1999.

[KR] R. Klette, A. Rosenfeld Digital Geometry, Morgan Kaufmann, San Francisco, 2004.

[L1] J.-C. Lin, "The Family of Universal Axes," Patt. Rec., Vol. 29, pp. 477-485, 1996.

[L2] J.-C. Lin, "Universal Principal Axes: An Easy-to-Construct Tool Useful in Defining Shape Orientations for Almost Every Kind of Shape," Patt. Rec., Vol. 26, pp. 485–493, 1993.

[LS] D.R. Lee, T. Sallee, A method of measuring shape, Geographical Review, Vol. 60, No. 4, pp. 555-563, 1970.

[LWL] Lee, S.C., Y. Wang, Lee, E.T., Compactness measure of digital shapes, Region 5 Conference: Annual Technical and Leadership Workshop, pp. 103- 105, April 2004.

[M] G. Marola, "On the Detection of Axes of Symmetry of Symmetric and Almost Symmetric Planar Images," IEEE Trans. PAMI, Vol. 11, No. 6, pp. 104-108, 1989.

[MS] M. Marji, P. Siy, Polygonal representation of digital planar curves through dominant point detection-a nonparametric algorithm, Pattern Recognition, Vol. 37, pp. 2113-2130, 2004.

[MSa] R.R. Martin, P.C. Stephenson, "Putting Objects into Boxes," Computer Aided Design, Vol.20, pp. 506-514, 1988.

[NB] O. Novaski and C. Barczak, Utilization of Voronoi diagrams for circularity algorithms, Precision Engineering Vol. 20, pp. 188-195, 1997.

[NC] V. Ng, and David W. Cheung, Measuring Asymmetries of Skin Lesions. Proc. IEEE International Conference On System, Man and Cybernetics (SMC-97), Orlando, Florida, USA, October, 1997.

[OLC] E. Orady, S. Li, Y. Chen, Evaluation of minimum zone straightness by a nonlinear optimization method, Journal of Manufacturing Science and Engineering, Vol. 122, pp. 795-797, Nov. 2000.

[OZ] P. O'Leary, P. Zsombor-Murray, Direct and specific least square fitting of hyperbolae and ellipses, Journal of Electronic Imaging, Vol 3, No. 13, pp. 492-503, July 2004.

[P] D. Proffitt, "The measurement of circularity and ellipticity on a digital grid", Pattern Recognition, Vol. 15, No. 5, pp. 383-387, 1982.

[Pa] S.E. Palmer, Vision Science: Photons to Phenomenology, MIT Press, 1999.

[PI] M. Peura, J. Iivarinen, Efficiency of simple shape descriptors, in: Advances in Visual Form Analysis (C. Arcelli, L.P. Cordella, G. Sanniti di Baja, eds.), World Sci., pp. 443–451, 1997.

[PB] Parabola, http://en.wikipedia.org/wiki/Parabola, 2008.

[PY] V. Shiv Naga Prasad and B. Yegnanarayana, "Finding Axes of Symmetry from Potential Fields," IEEE Trans. Image Processing, Vol. 13, No. 12, pp. 1559-1556, 2004.

[R1] P.L. Rosin, Measuring sigmoidality, Proc. CAIP 2003, LNCS 2756, pp. 410-417, 2003.

[R2] P. Rosin, "Measuring Rectangularity," Machine Vision and Applications, Vol. 11, pp. 191-196, 1999.

[R3] P. Rosin, "Measuring shape: ellipticity, rectangularity, and triangularity", Machine Vision and Applications, Vol. 14, pp. 172-184, 2003.

[R4] P.L. Rosin, "Techniques for Assessing Polygonal Approximations of Curves," IEEE Trans. PAMI, Vol. 19, No. 6, pp. 659-666, 1997.

[R5] P.L. Rosin, Ellipse fitting using orthogonal hyperbolae and Stirling's oval, CVGIP: Graph Models Image Process. Vol. 60, No. 3, pp. 209-213, 1998.

[R6] P.L. Rosin, Further five-point fit ellipse fitting, CVGIP: Graph Models Image Proc. Vol. 61, No. 5, pp. 245-259, 1999.

[R7] P.L. Rosin, Assessing error of fit functions for ellipses, CVGIP: Graph Models Image Process, Vol. 58, No. 5, pp. 494-502, 1996.

[RR] B. Ray, S. Ray, An algorithm for polygonal approximation of digitized curves, Pattern Recognition Letters, Vol. 13, pp. 489-496, 1992.

[RSV] I. Rothe, H. Süße, K. Voss, the method of normalization to determine invariants, IEEE T-PAMI, Vol. 18, No. 4, pp. 366-376, 1996.

[RZ] P. L. Rosin, J. Zunic, 2D shape measures for computer vision, in: Handbook of Applied Algorithms: Solving Scientific, Engineering and Practical problems, Wiley, pp. 347-371, 2007.

[S] H.I. Stern, Polygonal Entropy: A Convexity Measure, Pattern Recognition Letters, Vol. 10, pp. 229-235, 1989.

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

[SI] D. Shen, H.H.S. Ip, "Generalized Affine Invariant Normalization," IEEE Trans. PAMI, Vol. 19, No. 5, pp. 431-440, 1997.

[SICT] D. Shen, H.H.S. Ip, K.K.T. Cheung, and E.K. Teoh, "Symmetry Detection by Generalized Complex (GC) Moments: A Close-Form Solution," IEEE Trans. PAMI, Vol. 21, No. 5, pp. 466–476, 1999.

[SNS] N. Sladoje, I. Nyström, and P.K. Saha, "Measurements of digitized objects with fuzzy borders in 2D and 3D", Image and Vision Computing, Vol. 23, pp 123-132, 2005.

[SNS-2] N. Sladoje, I. Nyström, and P.K. Saha, "Measuring perimeter and area in low resolution images using a fuzzy approach", In Josef Bigun and Tomas Gustafsson, editors, Proceedings of 13th Scandinavian Conference on Image Analysis (SCIA 2003), Göteborg, Sweden, Lecture Notes in Computer Science, Vol. 2749, pp. 853-860, Springer-Verlag, 2003.

[SS] S. Singh, Shape Detection Using Gradient Features for Handwritten Character Recognition, Proc. 13th International Conference on Pattern Recognition (ICPR'96), Vienna, Austria, Vol. III, pp. 145-149, August, 1996.

[TC] W.H. Tsai, S.L. Chou, "Detection of Generalized Principal Axes in Rotationally Symmetric Shapes," Pattern Recognition, Vol. 24, pp. 95-104, 1991.

[TCa] J. Thureson, S. Carlsson, "Finding Object Categories in Cluttered Images Using Minimal Shape Prototypes", *13th Scandinavian Conference on Image Analysis SCIA,* Goteborg, Sweden, pp. 1122-1129, 2003.

[TS] A. Taza and C.Y. Suen, "Discrimination of Planar Shapes Using Shape Matrices," IEEE Trans. System, Man and Cybernetics, Vol. 19, No. 5, pp. 1281–1289, 1989.

[TR] V. M. Tuset, Paul L. Rosin, Antonio Lombarte, Sagittal otolith shape used in the identification of fishes of the genus Serranus, Fisheries Research, Vol. 81, pp. 316–325 2006.

[VS] Voss, K., Süße, H., Invariant fitting of planar objects by primitives. IEEE Trans. Patt. Anal. Mach. Intell. Vol. 19 pp. 80–84, 1997.

[W] E. Weisstein, "Eccentricity", MathWorld, http://mathworld.wolfram.com/Eccentricity.html, CRC Press LLC, 1999.

[YSN] Yu X., Z. Shen and S. Ninomiya, Measuring Geometrical Features of Insect Specimens using Image Analysis. Proc. 3rd AFITA;Asian Agricultural Information Technology and Management. Ed. F. Mei, pp. 591-595, 2002.

[Z] J. Zunic: "Boundary Based Orientation of Polygonal Shapes," Lecture Notes in Computer Science – PSIVT, Hsinchu, Taiwan, pp. 108-117, December 2006.

[ZKF] J. Zunic, L. Kopanja, J.E. Fieldsend, "Notes on Shape Orientation where the Standard Method Does not Work," Pattern Recognition, Vol. 39, No. 5, pp. 856-865, 2006.

[ZL] S.C. Zhang, Z.Q. Liu, A robust, real time ellipse detector, Pattern Recognition, Vol. 38, pp. 273-287, 2005.

[ZR1] Jovisa Zunic and Paul L. Rosin, Rectilinearity Measurements for Polygons, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 25, No. 9, pp. 1193-1200, 2003.

[ZR2] Jovisa Zunic and Paul L. Rosin, A new convexity measure for polygons, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 26, No. 7, pp. 923-934, 2004.

[ZRK] J. Zunic, P.L. Rosin, L. Kopanja: "On the Orientability of Shapes," IEEE Transactions on Image Processing, Vol. 15, No. 11, pp. 3478-3487, 2006.