MVA2009 IAPR Conference on Machine Vision Applications, May 20-22, 2009, Yokohama, JAPAN

Measuring conicity from shape boundaries

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

Abstract

There has been a lot of research on ellipse fitting and measuring ellipticity of a set of points. However, when the shape is primarily hyperbolic or parabolic, there are no existing methods to measure such properties. This paper 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 evaluation of fits transforms the point data into polar representation where the radius in this representation is equal to 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 linearity of the polar representation will correspond to the quality of the fit for the original data. The conicity measure is tested on a set of 45 shapes.

1 Introduction

Classifying a shape as a certain primitive is important in image processing applications and computer vision systems. Popular shape measures such as elongation, convexity and orientation exist in literature [3, 5, 8, 9]. Finding a shape that best represents a set of points is called fitting. Like other measures for primitive geometric shapes, the measures of conic fitting are motivated by real world image processing problems. Ellipticity is common in nature and industry, and finding a way of identifying it can be important to both. Applications of ellipse identification are found in agricultural and medical imaging systems for identifying certain grains, onions, watermelons, cells, and even human faces [4]. 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) [7]. In this article we study two related problems: conic fitting and measuring of shape ellipticity, hyperbolicity and parabolicity, via a unified notion of conicity. One of the main advantages of our algorithm is that it is boundary based and therefore works on both open and closed curves. Our algorithm has image processing applications in mind which deal with pixels as points, but also accept real numbers as input.

In designing our algorithm for conic fitting and measuring, we restrict ourselves to the following criteria. Conicity values are assigned to sets of points and these values shall be numbers in the range [0, 1]. The conicity measure equals 1 if and only if the shape is an ellipse, hyperbola or parabola, and is close to 0 when the shape is highly non conic. A shape's conicity value should be invariant under similarity transformations of the shape, such as scaling, rotation and translation. Conicity values should also be computed by a simple and fast shape based algorithm. As in existing literature [1, 2], the points in the set are not ordered; that is, conicity fits and conicity measures do not depend on the ordering of points in the data set or along the boundary.

Stojmenovic and Nayak [5] proposed a novel way of measuring the accuracy of ellipse fits against the original point set. The solution is to transform 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. They 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. They discuss two ways of determining the center of the shape.

Ellipse fitting has been widely studied in literature. Hyperbola fitting was studied in [1, 7], but they do not address the quality of the fit in these studies. The algorithm presented here fits a conic to a set of points, and evaluates the quality of the fit. We will apply a shape sampling based method [2] by Rosin because his papers indicate it as the ultimate selection. The conic fit is done by taking *n* quintuplets of points as proposed by [2]. 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. 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 [5]), 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 Polar angle becomes proportional to the complete range of angles [0, 2π], by stretching the original range of angles to fit the new one if necessary. The radius is multiplied by a factor to convert it to an angular range, so that the conicity remains the same regardless of the shape's scale. The linearity of the polar representation will correspond to the quality of the fit for the original data. Any one of the six linearity measures in [3] can be used as a base of our conicity algorithm. We chose the 'average orientations' linearity measure since it was previously shown to be most accurate compared to human perception.

Overall in this paper, we propose a novel algorithm, which combines the fit of point sets to conics as described in [2] with our own assignment of conicity values to sets of points in 2D. The detailed literature review is omitted due to space constraints, and can be found in [5]. The new measures are presented in section 2. Section 3 is reserved for the presentation of our test set along with a general discussion of our results.

2 Measuring Conicity: Ellipticity, Hyperbolicity and Parabolicity

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 [2] 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.

2.1 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 [5] 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.

2.2 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 3 illustrates this point.

We exploit inherent hyperbola property (see Figure 3) 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 4, 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.

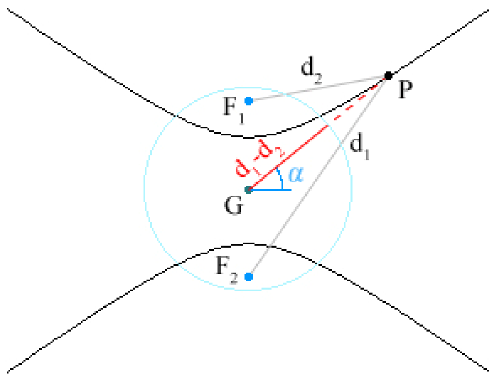


Figure 3 Measuring the hyperbolicity of the point set

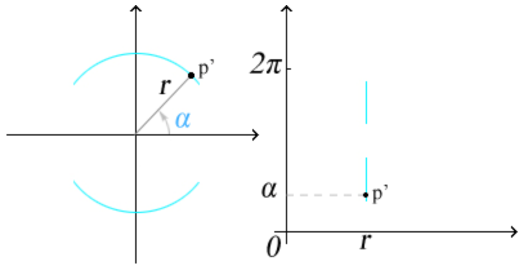


Figure 4 Polar point set of a hyperbolaon a planar graph

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 4. 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.

2.3 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 6 demonstrates the exploited parabola property used in our work. Figure 6 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

[6] 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.

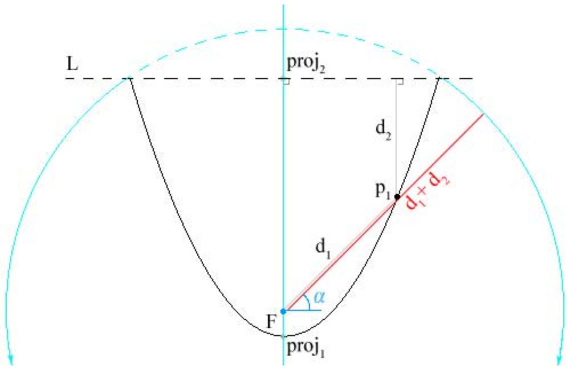


Figure 6 Measuring the parabolicity of the point set

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 section 3.1. 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 6, 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: $r'=(r\cdot q)/(|proj_1\text{-}proj_2|)$, 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 6. 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.

3 Experimental data and results

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.

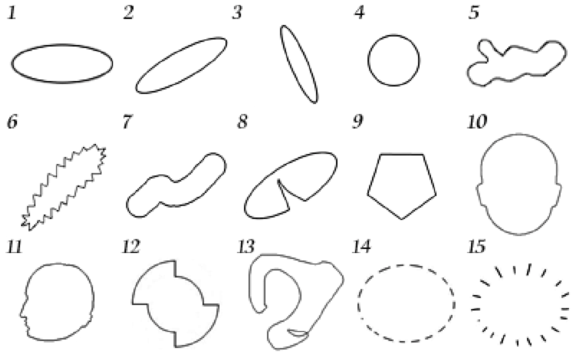
In practice, our experiments were performed on images where the shapes were represented by pixels with integer coordinates. As such, it is very difficult to obtain a perfect shape, but rather a close approximation. This restriction did not affect the ability of the algorithm to identify ellipses and 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: $b^2\text{-}4ac=0$, 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 $b^2\text{-}4ac=0$. 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 section 3.1.

The conicity algorithm was tested on a set of 45 shapes, shown in Figures 7, 8 and 9. 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.

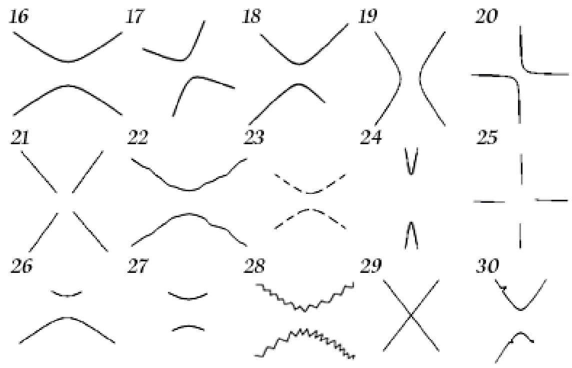
Table I shows the conicity values for all of the test curves. For each curve, we see the shape designation assigned to it by the algorithm in the 'detected' column, and its corresponding conicity value in the 'C' column. When these 2 descriptors are taken simultaneously, the algorithm classifies a shape as 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.

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 8, 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.

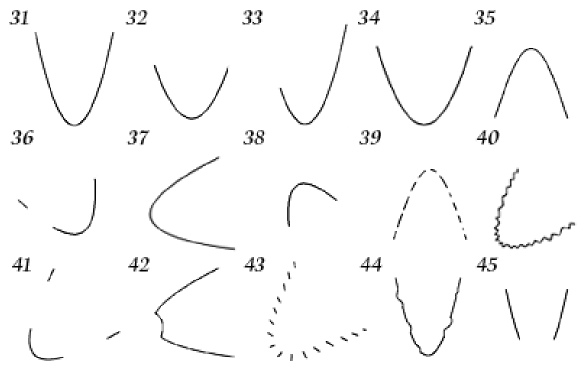
Shapes 16-30 were also correctly classified as hyperbolic by the algorithm. Hyperbolas are inherently comprised of disjoint point sets, but even when these disjoint sets are further broken up into smaller point sets, they can still be identified as hyperbolas, such as in the case of shape 23. Most of 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.



7. Elliptic shapes in test set



8. Hyperbolic shapes in test set



9. Parabolic shapes in test set

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.

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.

TABLE I. Conicity values of test shapes

#	detected	C	#	detected	C	#	detected	C
1	ellipse	.989	**16**	hyperbola	1.00	**31**	parabola	.999
2	ellipse	.978	**17**	hyperbola	.972	**32**	parabola	.980
3	ellipse	.976	**18**	hyperbola	.934	**33**	parabola	.993
4	ellipse	.917	**19**	hyperbola	.971	**34**	parabola	.998
5	ellipse	.778	**20**	hyperbola	.939	**35**	parabola	.966
6	ellipse	.897	**21**	hyperbola	.871	**36**	parabola	.962
7	ellipse	.896	**22**	hyperbola	.911	**37**	parabola	.984
8	ellipse	.938	**23**	hyperbola	.995	**38**	parabola	.973
9	ellipse	.632	**24**	hyperbola	.969	**39**	parabola	.945
10	ellipse	.931	**25**	hyperbola	.962	**40**	parabola	.880
11	ellipse	.800	**26**	hyperbola	.982	**41**	parabola	.940
12	ellipse	.573	**27**	hyperbola	.989	**42**	parabola	.936
13	ellipse	.593	**28**	hyperbola	.876	**43**	parabola	.718
14	ellipse	.933	**29**	hyperbola	.575	**44**	parabola	.932
15	ellipse	.698	**30**	hyperbola	.987	**45**	parabola	.936

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.

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.

4 References

1. F.L. Bookstein, Fitting conic sections to scattered data, Computer Graphics and Image Processing, 9, 56-71, 1979.
2. P.L. Rosin, Further five-point fit ellipse fitting, CVGIP: Graph Models Image Proc, 61, 5, 245-259, 1999.
3. Milos Stojmenovic, Amiya Nayak, Jovisa Zunic, Measuring Linearity of a Finite Set of Points, Pattern Recognition, Vol. 41, pp. 2503-2511, 2008.
4. Victor M. Tuset, Paul L. Rosin, Antonio Lombarte, Sagittal otolith shape used in the identification of fishes of the genus Serranus, Fisheries Research 81, 316–325, 2006.
5. M. Stojmenovic, A. Nayak, Direct Ellipse Fitting and Measuring Based on Shape Boundaries, Lecture Notes in Computer Science, Vol. 4872, pp. 221-235, 2007.
6. Parabola, http://en.wikipedia.org/wiki/Parabola, 2008.
7. 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.
8. M. Stojmenovic, J. Zunic, Measuring elongation from shape boundary, Journal of Mathematical Imaging and Vision, Vol. 30, No. 1, pp. 73-85, 2008.
9. J. Zunic, M. Stojmenovic, Boundary based shape orientation, Pattern Recognition, 41, 5, 1785-1798, 2008.